Implementing Finite Impulse Response (FIR) Low Pass Filter(LPF) with Different Number Representations

Miao Li

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Engineering
in
Electrical and Computer Engineering
at the
University of Canterbury,
Christchurch, New Zealand.

2016

ABSTRACT

Number representation can be used for representing the coefficients of the digital filter as a means of reducing the multiplication size and improved the computation speed. However, when each coefficient is rounded to the different number representations, their quantization different error is caused. This quantization round-off error of coefficients can influence the magnitude of the stopband attenuation when implementing the finite impulse response(FIR) low pass filter(LPF). The number representation systems here include two's complement number representation system, canonical signed digit(CSD) number representation system and sum of power-of-two(SPT) number representation system.

In this work, we analyze the round-off error of coefficient of digital filter using different number representation systems and give the probability density distribution of round-off error at various word-lengths. As the SPT number representation is also related to the Hamming weight K, the probability density distribution changes with varies the value of the K. Then implementing the FIR LPF filter with the different number system to find out the influence of coefficients quantization on the stopband attenuation.

Furthermore, a cost function is used to connect the computation size and filter performance together to find a FIR LPF which has acceptable performance and quicker computation. This cost function is used to indicate the proper word-length and filter length for approximate FIR LPF which achieved by different number representations systems. After comparison of 1159 of approximate FIR LPF used different number representation, we try to find out the suitable number representation which can make the approximate filter has better filter performance and lowest computation size.

ACKNOWLEDGEMENTS

My study of Master degree at the University of Canterbury has been full of challenging. I have to thank many people who has helped me during my research.

First of all, I would like to thank my supervisor, Dr Andrew Bainbridge-Smith for his guidance, patience and supports. During my research, he gave me a lot of suggestions and confidence and took time out of his busy schedule to help me with my thesis.

Secondly, I would like to express my thanks to my friend, Victor Wang, who patiently helped with my thesis and also gave me many supports and useful suggestions. I would also like to thank to all the friends in New Zealand, they made my life full of surprise and excitement.

Lastly, I have to thanks my parents. They gave me the motivation and supports all the way. Because of them, I feel my life is more enjoyable and happily during all these time.

Miao Li

University of Canterbury

2016

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ASIC	application specific integrated circuit
CSD	canonic signed digit
FPGA	field programmable gate array.
FIR	finite impulse response.
LIT	linear, time-invariant.
LPF	low-pass filter.
IIR	infinite impulse response.
RBR	redundant binary representation .
SPT	sum of power of two.
SMPT_K	K terms of signed magnitude sum of power of two.
SPT_K	K terms of signed sum of power of two.
SPTCSD_K	K terms of signed sum of power of two with CSD constraint
SD	signed digit.
WHD	minimum hamming weight.

Chapter 1

INTRODUCTION

Digital signal processing has developed rapidly in past 50 years. It has brought a lot of significant achievements and advances to our engineering and science fields, such as communication, audio system, imagine compression, antenna systems and speech processing [1, 2, 4]. With the development of microelectronic integrated hardware, digital signal processing also experienced a revolution. The smaller, faster and cheaper integrated-circuit have made the digital signal processing hardware not only more computation powerfully but also more compact, reliable and sophisticated as well as inexpensive [1]. However, because the analog to digital converters still cannot work fast enough and the computation is too complex to performed in real-time high frequency signals are different to process efficiently. To solve these problems, we still need make every effort to develop digital signal processing.

Digital filter are the foundation of digital signal processing. The design of filters is the basic unit in all the digital processing application. In the past 50 years, different kinds of filters have been designed, becoming more accurate and operating at higher speeds. Research on the method of design for smaller and more effective Finite Impulse Response (FIR) filter have been experienced by generations [5].

One method to speed up the computation is implementing FIR filter using appropriate number representation [6]. As we know that numerical values can be represented in many different ways, like non-positional Roman Numerals or positional Hindu-Arabic Numerals (0,1,2,3...9) [7, 8]. The positional number system is more widely used because it simplifies a number of arithmetic operations. In this way, the number representation used in the FIR filter is to replace the traditional representation with more appropriate representation for coefficients as to simplify

the computation [9].

1.1 RESEARCH OBJECTIVES

In this thesis, we analyze the coefficients quantization error effects on the frequency response of finite impulse response(FIR) low pass filter(LPF) used different number representation systems and implementing FIR LPF using these number representations. This thesis includes:

- A theoretical analysis on coefficient quantization errors of using different number representations.
- Implementing FIR LPF with different number representation systems.
- A cost function is proposed to connect the filter performance and computation together. This cost function is used to indicate the proper word-length and filter length for implementing FIR LPF filter with different number representations.
- A comparison of cost for each number representation system.

1.2 OUTLINE OF THESIS

The organization of the thesis is as follows:

Chapter 2 describes the background of digital filter and some main factors related to the FIR filter design. These factors include the filter specifications, basic computation cost, word length, filter length and quantization effects.

Chapter 3 introduces the four number representation systems. There are common binary number representation, redundant number system, canonical signed digit(CSD) number representation and sum of power-of-two number representation.

Chapter 4 provides the coefficients quantization error of each number representation and gives the experimental and simulation results of implementing the FIR LPF with different number

representations. This chapter gives the cases of probability density distribution for SPT number representation system at various word-length and their distribution trend.

Chapter 5 gives a cost function to measure the filter performance and computation size. This cost function helps to choose the proper word-length and filter length for approximate FIR LPF filters. It compares the cost of FIR LPFs using the CSD number representation and using SPT number representation when the value of K increases.

Chapter 6 concludes the research and future word.

Chapter 2

BACKGROUND

In the digital signal processing field, digital filters play an import role and are widely used in the language of signal processing, image signal processing, biomedical signal processing, and other areas. A digital filter is a mathematical operations performed on a digital input signal to reduce unexpected signals or enhance some desired signals [4]. Digital filters are more accurate, more reliable, and easier to integrate due to being programmable [1].

Digital filters can be classified as time-invariant or time-varying, linear or non-linear [4, 5, 10].A linear system is shown in Figure 2.1. This system is also a simple Finite Impulse Response(FIR) filter. In this thesis, we focus on the design and implementation of linear and time-invariant (LTI) finite impulse response (FIR) filters.

2.1 SPECIFICATION OF DIGITAL FILTER

As mentioned, digital filters have number of advantages over analog filter [2, 11]. When implementing a digital filter it is necessary to consider its specification. Figure 2.2 shows the important characteristics of a low-pass digital filter. The passband, transition band and stopband regions, passband ripple and stopband attenuation are all illustrated. Passband is a range of frequencies that pass through a filter with little attenuation. In Figure 2.2, the passband

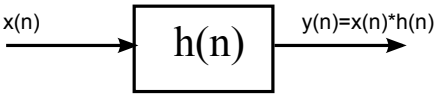


Figure 2.1: A simple linear system expressed in time domain.

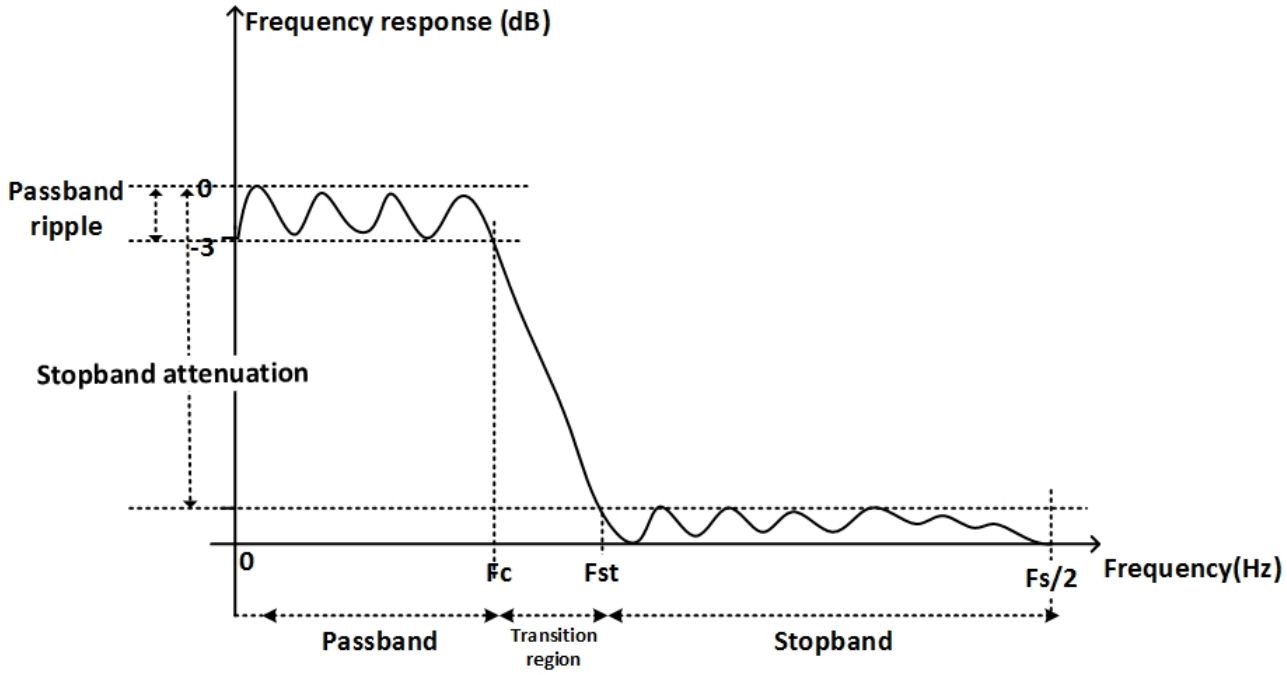


Figure 2.2: A low-pass digital filter frequency response [1].

indicates the frequency region is greater than -3 dB. The stopband is the region where there is full attenuation. The transition region is the band of frequencies between the passband and stopband. Passband ripple refers to fluctuation in the passband and is measured in dB. Stopband attenuation is measured between the peak passband amplitude and the largest stopband lobe amplitude [1, 2, 12].

Other filter specification such as cut-off frequency, overshoot and rolloff may also be considered in digital filter design. The cut-off frequency is the frequency at which the ratio of the (input/output) has a magnitude of 0.707, which is -3 dB. Rolloff is used to describe the slope of the filter response between stopband and passband, usually a higher roll-off rate is close to ideal. Overshoot is presented when the output of a filter has a larger value than the input, especially for the step response [2, 13].

Besides, the filter length and word-length are also have the crucial influence on the digital filter design. The filter length is the number of coefficients. The word-length is the number of bits used to quantize the coefficients and signal values [4]. These details are covered in Section 2.2.3 and Section 2.2.4.

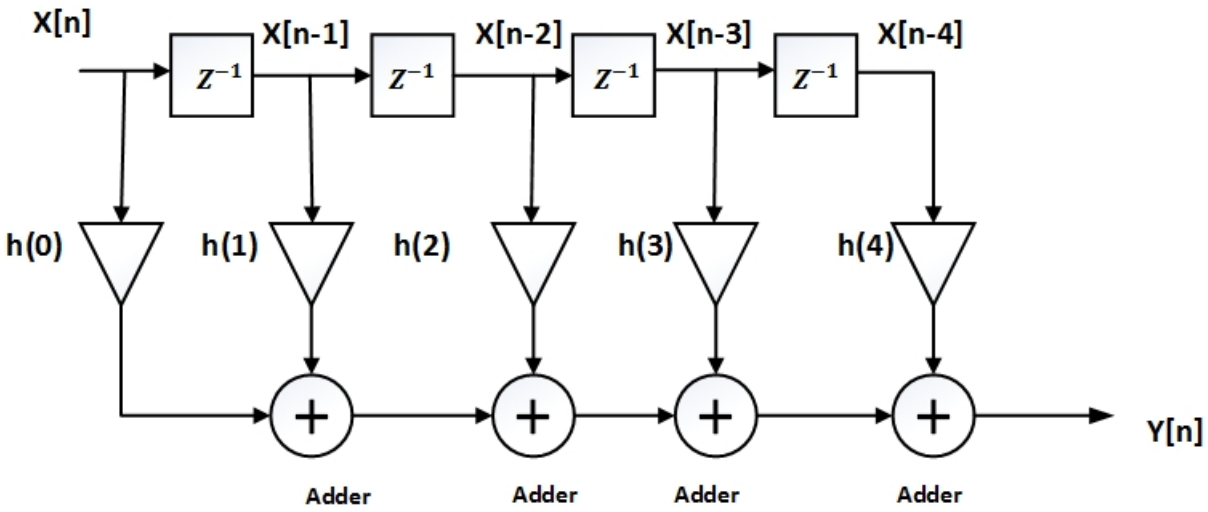


Figure 2.3: Direct form FIR digital filter structure [1].

2.2 FIR FILTER

An FIR filter is a digital filter whose impulse response is a finite-duration [5]. If the input of the FIR filter becomes a sequence of zeros, its output will also become zeros [14]. The no feedback structure of FIR filter characteristic make the FIR filter inherently stable [15]. The lack of phase distortion is another advantage of the FIR filter. This leads to the FIR filter can be linear phase filter [2].

A direct form discrete-time FIR filter of order N is [1, 4, 11]

$$y[n] = h_0x[n] + h_1x[n-1] + \cdots h_Nx(n-N) = \sum_{i=0}^{N} h_ix(n-i), \tag{2.1}$$

where $x[n]$ is the input signal while $y[n]$ is output signal. h_0,h_2,...,h_N are the coefficients of the FIR filter. Figure 2.3 is a direct form discrete-time FIR filter of order 4. Z^{-1} is the an N-stage delay line.

2.2.1 Pole-zero Plots

A pole-zero plot shows the location (in the z-plane) of the poles and zeros in a dynamic way [1, 2, 15]. It is a useful graphical representation for conveying some properties of the filter in

frequency domain and z-domain, such as stable or minimum phase. In z-plane, there is a unit circle which has the radius is 1 [15]. The transfer function $H(z)$ is the as [2, 12, 15],

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^{N} b_n z^{-n}}{1 + \sum_{n=0}^{M} a_m z^{-m}} \tag{2.2}$$

where the B and A are polynomials in z. The zeros are roots of the $B(z)$ and the poles are the roots of the $A(z)$. A plot contains zeros and poles of a system on a z-plane is the Pole-zero plots. 'o'(circle) is used to represent the zeros and 'x'(cross) refers to the poles in a Pole-zero plot [15]. Pole-zero plots can examine the frequency domain and z-domain conveniently.

The placement of poles and zeros are based on mapping of frequencies to the z-domain, the design of filter using pole-zero method relates to trail and error [2]. The locations of poles and zeros work with each other to give a impact on the response. Usually, the higher filter order and more poles and zeros, the easier to meet the ideal specifications [2, 4].

As mentioned previously, an FIR filter has two properties which are stable and linear phase. FIR filters are always stable because poles are always within the unit circle, so the poles cannot influence the stability of an FIR system [2]. Figure 2.4 illustrates that the pole-zero plot works as FIR LPF. The zero is located at the origin of the unity circle and pole is at the positive axis but still inside unit circle. The vector length of numerator (the distance from zero to unit circle) is always 1 but the denominator (the distance from pole to unit circle) increases with increasing of frequency (from position A to position B). In this way, the magnitude decreases (the ratio of the numerator and denominator lengths) which works as a LPF frequency response [2].

2.2.2 Multiplication Size

Registers, adders and multipliers are essential elements for implementing a FIR filter in hardware [16]. Figure 2.5 shows the delay, multiplier and adder in FIR filter.

The implementation can be described as the following procedure [16]:

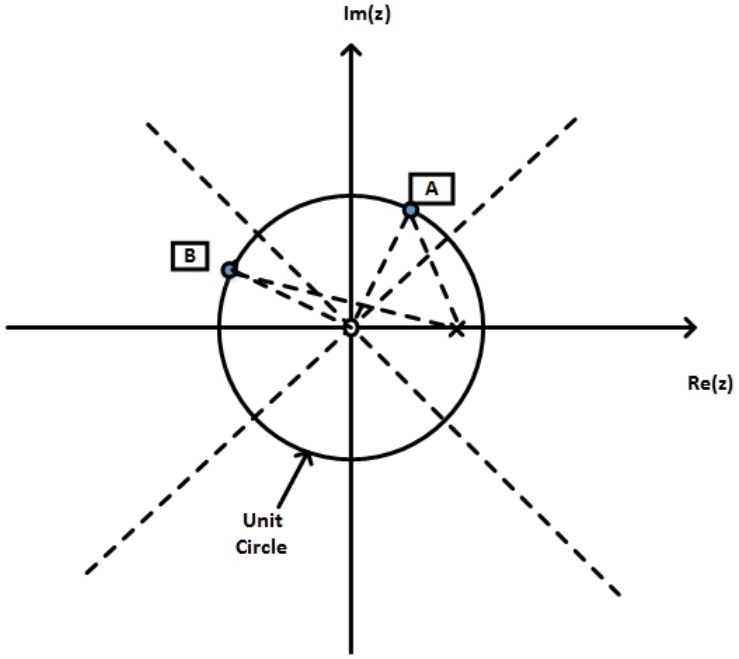


Figure 2.4: Pole zero plot of a low pass filter [2]

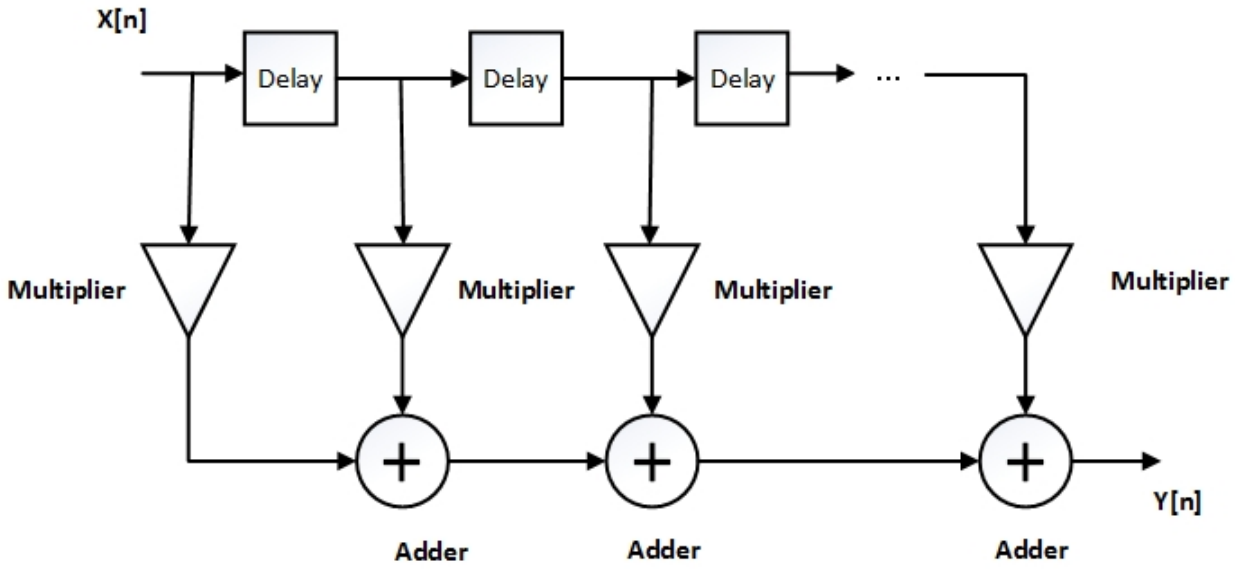


Figure 2.5: Delay,multiplier and adder in FIR filter structure

1. Feed the input signal sample into the delay line.

2. Multiply each sample by the corresponding coefficients in the delay line and accumulate the result. Usually, it is called MAC operation.

3. Shift the delay line by one sample to make room for the next input sample.

Multiplication size we discussed here is the number of multipliers which is used in MAC operations. Multiplication size can be reduced by eliminating the multipliers, such as splitting multipliers or using some number representation systems.

2.2.3 Effects of Finite Word-length

In practice, digital signals and filter coefficients are represented by the finite number of bits, this results the quantization effects. Quantization is commonly performed by approximating values using a fixed number of bits, where there are various representations and methods of rounding. Quantization leads to rippling in the stopband response such that the performance differ from the digital design specification [2]. Figure 2.6 shows the response of a FIR LPF with 32 coefficients using a word-length of 8 bits, and Figure 2.7 depicts the same filter except with 12-bits words. The filter of Figure 2.7 is clearly a better approximate to the original infinite-precision filter than the filter in Figure 2.7, particularly in the stopband behavior.

The use of limited word-length is an important characteristic of practical digital filter implementation. Word-length affects the quantization noise in the coefficients, as well as having implications on aspects of the hardware implementation (memory, computation time, etc).

The poles and zeros will differ from the original poles and zeros after quantization, resulting in a different frequency response. There are four main problems that may result from using a finite word-length [1, 2, 4, 14, 17]:

- Quantization noise, which is controlled by signal- to-noise ratio. Increasing number of bits can improve SNR. Oversampling is another way to reduce it.

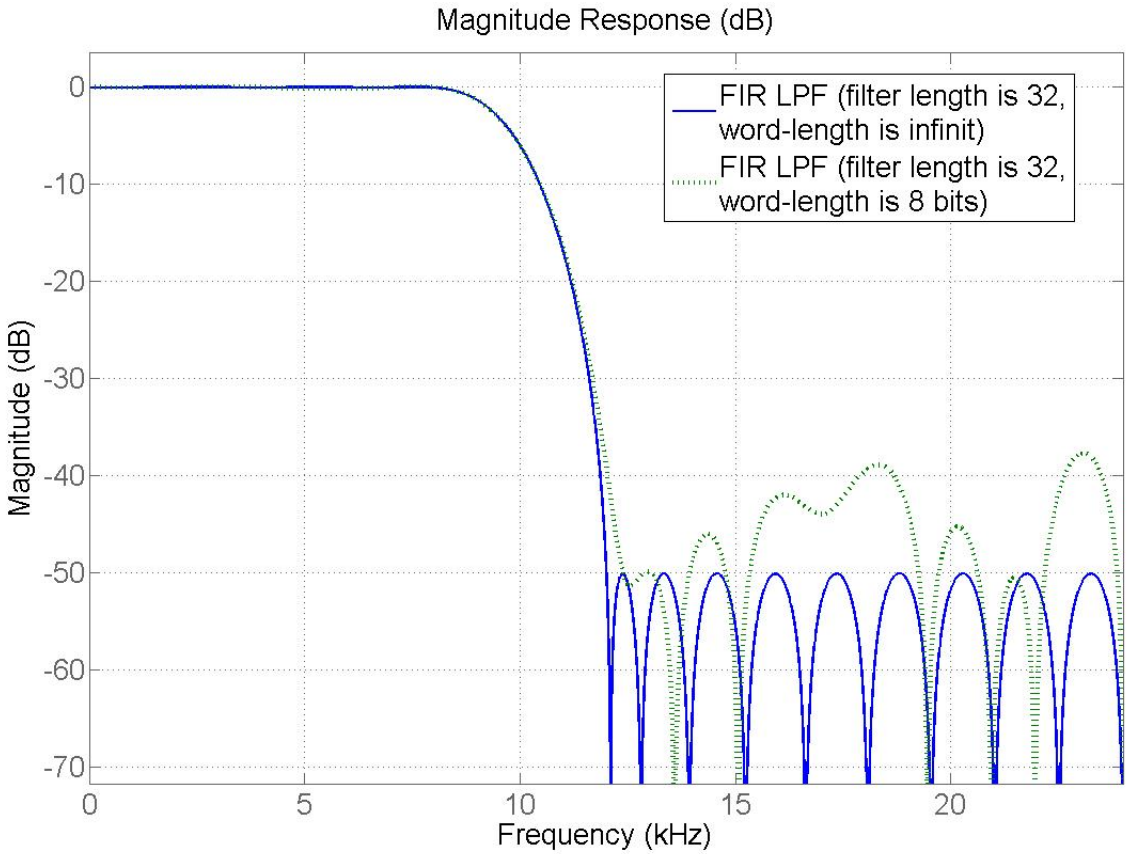


Figure 2.6: The frequency response of 31 order FIR LPF with coefficients quantized at 8 bits

- Coefficient quantization, which as we mentioned before, quantized coefficient cause frequency response changing. For example, a larger passband ripple or a smaller stopband attenuation.

- Roundoff errors, which refer to store result of a multiplication after lower-order bits have already been discarded. In this way, there is a roundoff error. This error can be controlled by different arithmetic and filter structure.

- Overflow, which is caused by arithmetic operations. Sometimes, the result of sum of two large number will exceeds the initial word-length. It is necessary scale coefficients of filter in case overflow.

The coefficient quantization will be the one of our research objectives in our thesis.

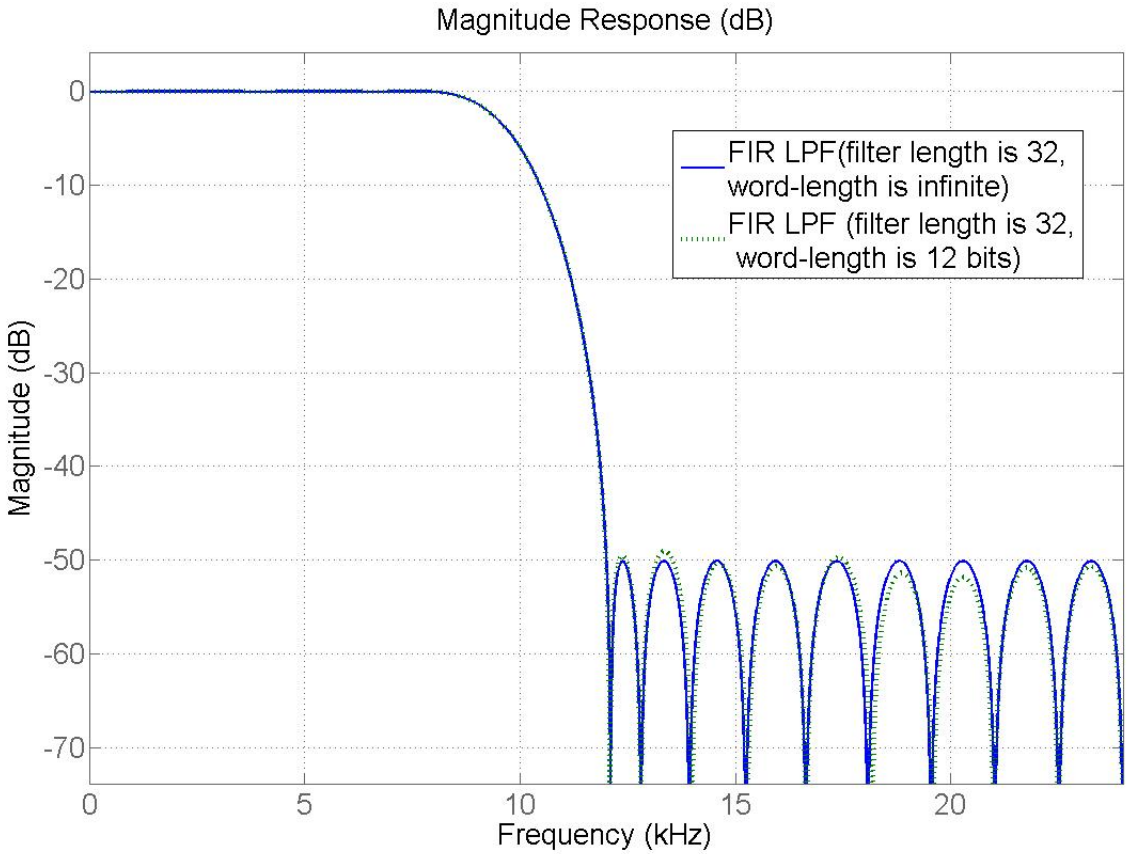


Figure 2.7: The frequency response of 31 order FIR LPF with coefficients quantized at 12 bits

2.2.4 Filter Length

Filter length is the number of coefficients. Larger filters tend to have better performance, as roll-off becomes sharper and the passband ripples and stopband ripples are reduced [2, 4, 5, 14]. By way of example the frequency response of three FIR filter (with filter length is 32, 42 and 52) are plotted, Figure 2.8 with the desired passband and stopband frequencies at 8 KHz and 12 KHz respectively. It is shown that the performance of a filter improves with greater filter length. Compared with the FIR filter which filter length is 32, FIR filter with 52 filter length obtains sharper roll-off and better stopband attenuation. However, the use of large filter lengths requires more computation when convolving the input signal with the filter coefficients [2]. In this way, the filter length also gives a influence on the multiplication size when implementing a filter on the hardware.

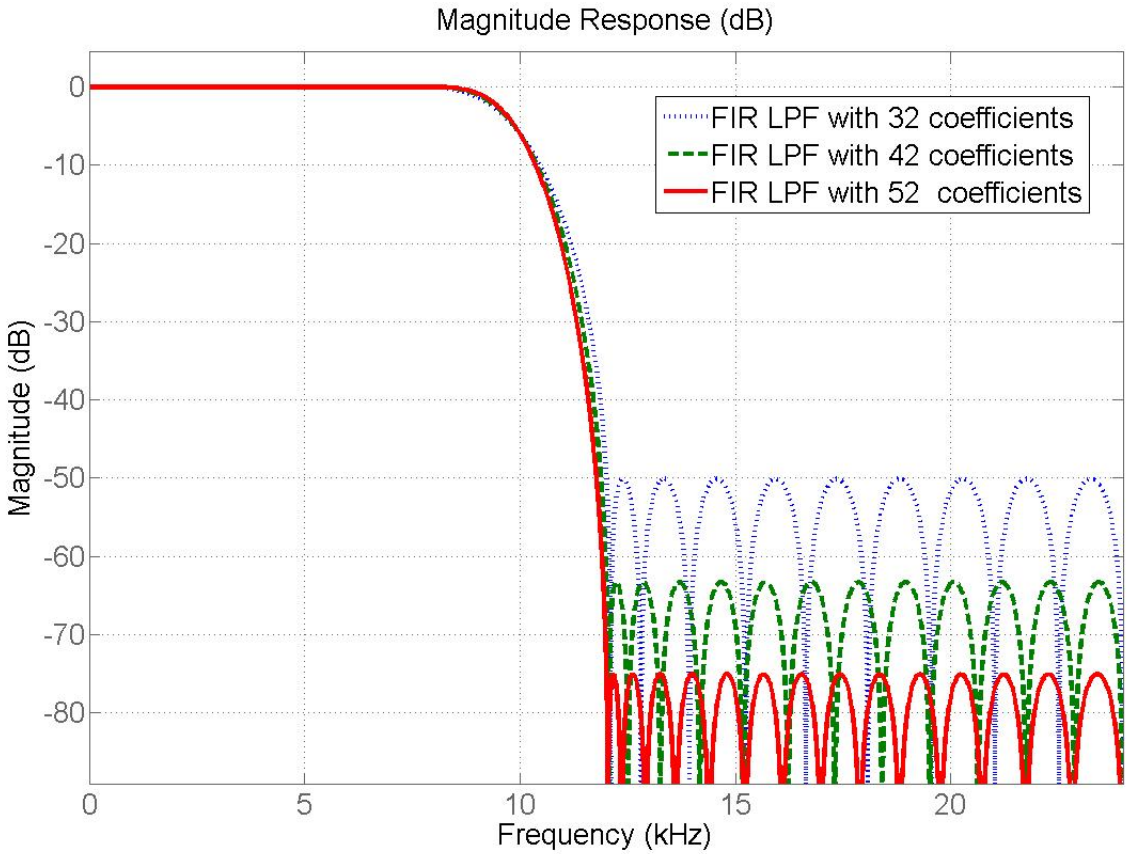


Figure 2.8: The frequency response of FIR Filter with different filter length

2.2.5 Quantization

Nearly all digital signal processing involve quantization. Quantization is the process of converting continuous values to a relatively discrete set [2]. If each sample is quantized as a set of zeros and ones during quantization, it is called digitally quantized. Quantization causes noise and loss of information of signals [1, 2, 4], which is the quantization noise. The noise caused by three main source [6, 18]. There are input quantization [19, 20], coefficient quantization [17, 21–23] and quantization caused in arithmetic operations [24]. The quantizations error can be caused by rounding, truncation and sign-magnitude truncation. The error between an input value x and its quantized value $Q_r(x)$ is the quantization error e_r, which is described by Equation 2.3. The quantization error discussed here is round-off error. Hence, the round-off quantized error is symmetric about zero [2, 17].

$$e_r = Q_r(x) - x \tag{2.3}$$

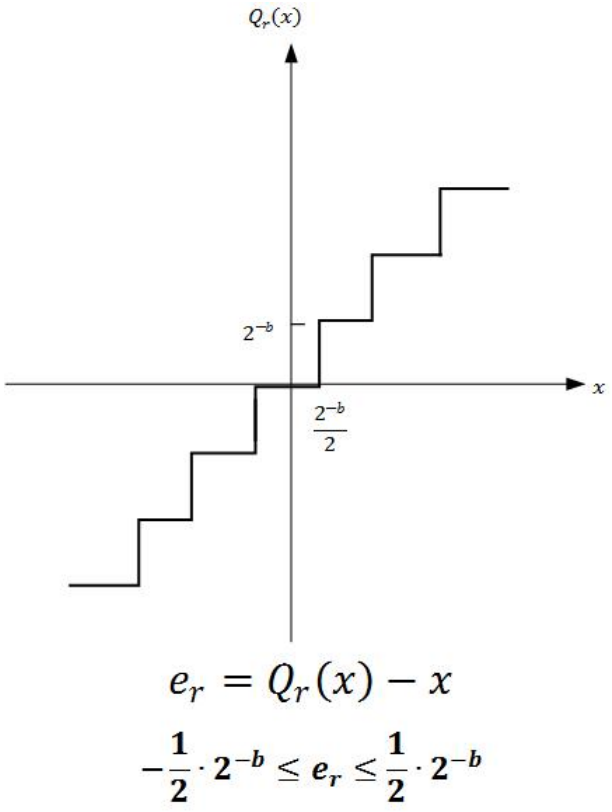


Figure 2.9: Quantization error in rounding [1].

Figure 2.9 demonstrates when x is a continuous-valued signal amplitude. A signal $x[n]$ could fall into the full-scale range which is its maximum variation $D = x_{max} - x_{min}$. If $x[n]$ is quantized to L quantization levels with the full-scale range of D, then the resolution (also called quantization step size Δ) is,

$$\Delta = D/L \quad . \tag{2.4}$$

For an 8-bit quantizer, the range is $20 \log 2^8 \approx 48$ dB [2]. For rounding quantization of simple two's complement, when L become infinite, the error value can equally distribute at any position in the range $-\Delta/2$ and $\Delta/2$, which is a uniform distribution [1, 2]. Figure 2.10 illustrates when input value quantized to two's complement values, the probability density distribution of round-off error strictly obey the uniform distribution. The quantization step size is 2^{-b}.

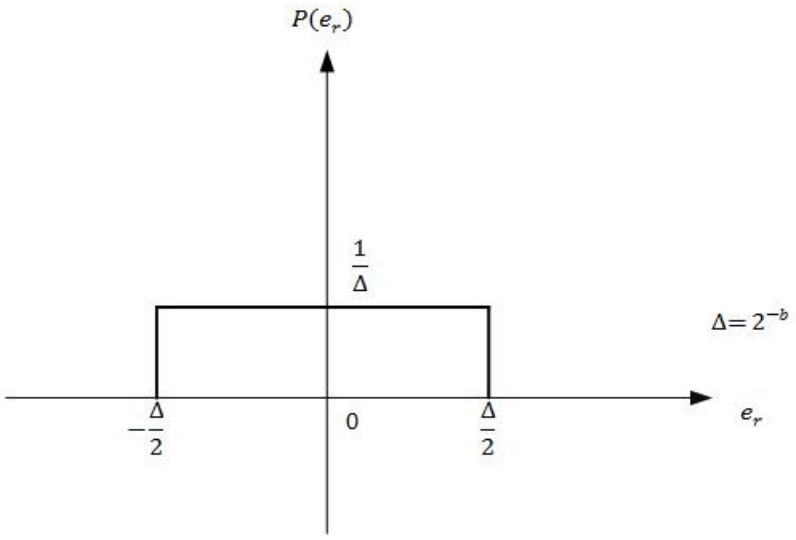


Figure 2.10: Statistical characterization of round-off quantization errors

2.2.6 Quantization of Filter Coefficients

For FIR filter, the coefficients quantization has the influence on the magnitude. Figure 2.11 gives the comparison of frequency response of the 31 order FIR LPF with the quantized coefficients and unquantized coefficients. It is obviously that the coefficients quantization affects on the magnitude of the stopband ripples.

When representing coefficients of digital filter, a moderate word-length is considered to keep the reduce the error in the frequency response. When quantizing a coefficient to $(b+1)$ bits, if the filter length increase, then the word-length also should be increased to keep the same round-off error. The error of rounding the coefficient also follows the uniform distribution [1,17]. For a M order filter, the round-off error frequency response can be,

$$E_\omega = \sum_{n=0}^{M-1} e(n)e^{-jwn}, \tag{2.5}$$

where the $e(n)$ is the quantized error. As the error also follows the uniform distribution, so the variances of the error E_ω is [1],

$$\delta_E^2 = \frac{2^{-2(b+1)}}{12}N = \frac{2^{-2(b+2)}}{3}N \quad , \tag{2.6}$$

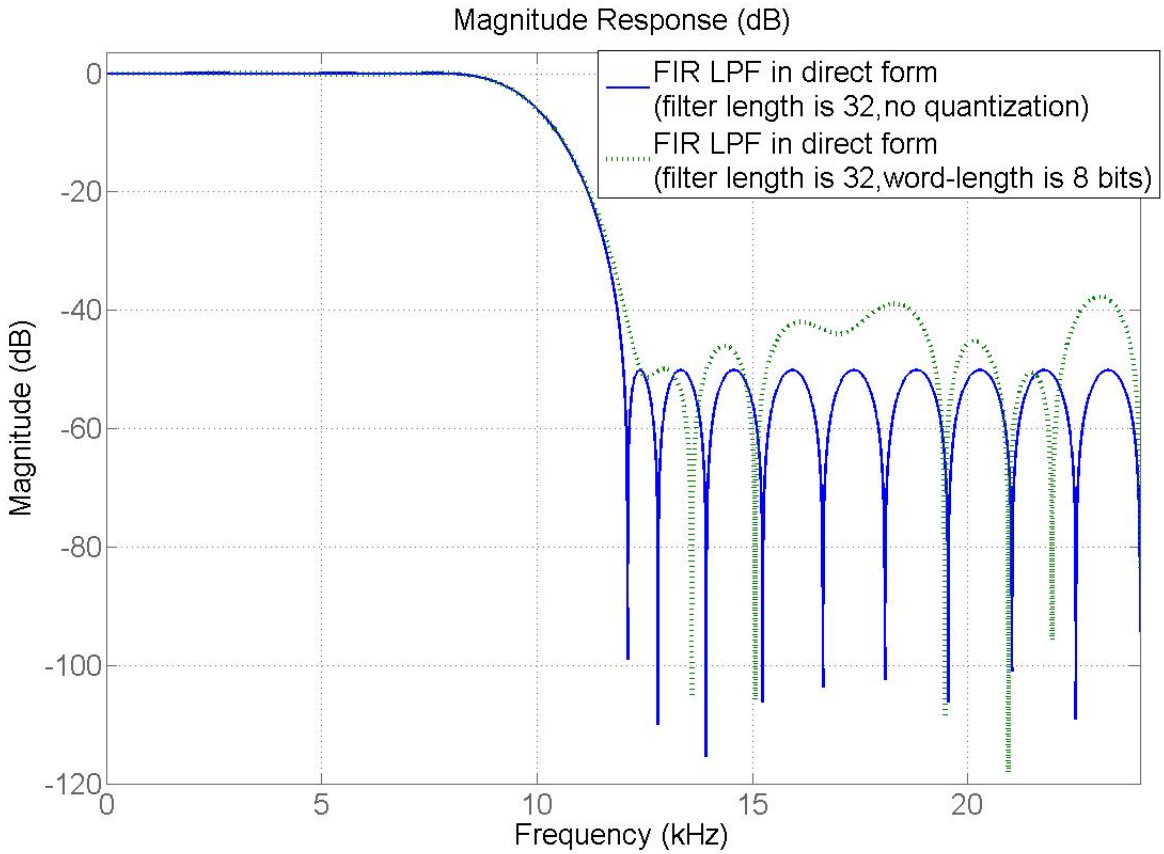


Figure 2.11: Effect of coefficients quantization of an 31 order FIR LPF

and its standard deviation is [1],

$$\delta_E = \frac{2^{-2(b+2)}}{\sqrt{3}}\sqrt{N} \quad . \tag{2.7}$$

From above stand deviation, it is clearly that in order to keep the δ remains 0, the word-length has to increase extra 1 bit with increasing every factor of 4 filter length. In this way, the effect of the magnitude on the frequency response can be fixed at the same level.

2.3 SUMMARY

In this chapter, we give general background informations of the FIR digital filter. It contains the specifications of the filters and the other factors related to the filter design. The specification of filter including the passband edged frequency, stopband edge frequency, the worst stopband attenuation and the filter length, some high demanded filter also requires the sharp of the

response in some specified frequency band.

The pole and zero plots is presented to help understanding the frequency response in the z-domain. It is also introduced filter length and word-length gives the impacts on the filter performance. The filter length influences the multiplication size and the finite word-length gives coefficients quantization effects which will affect on the magnitude of frequency response.

Chapter 3

NUMBER REPRESENTATION SYSTEM

The main purpose of this chapter is to introduce the different kinds of number representation systems. The common binary number representation, redundant number representation, canonical signed digit(CSD) number representation and sum of power two (SPT) number representation are detailed. Each number representation scheme has its advantages, we specifically look at the pseudo-redundant number representations to improve computational cost.

3.1 COMMON BINARY NUMBER REPRESENTATION

3.1.1 Unsigned Binary Number Representation

The unsigned binary number system is a commonly used representation in digital systems. Unsigned binary is a positional number system with base two, where digits are either zero(0) or one(1) [9]. Each digit is referred to as a bit [2, 25]. A whole number X can be represented using n binary digits according to,

$$X = \sum_{i=0}^{N-1} \alpha_i 2^i \quad \alpha_i \in \{0, 1\}. \tag{3.1}$$

For example, the number $X = 149_{10}$ can be express as 8 bits, $\alpha = 10010101_2$. The range of whole numbers that can represented by an N-bit unsigned binary number is

$$0 \leq X \leq 2^N - 1. \tag{3.2}$$

Hence, a one byte (8 bit) number can represented 0 to 255.

3.1.2 Signed Binary Number Representation

Signed Binary is also base two but is able to encode negative integers as well as positive integers. The two common methods of signed binary representation are described here: signed magnitude and two's complement [25].

3.1.2.1 Signed Magnitude Number Representation

Signed magnitude representation allocates a " sign bit" in sequences of the binary set. Typically, the most significant bit is designated as the sign bit, where a sign bit of 1 indicates a negative sign and a zero sign bit indicates position sign [3, 7, 26]. Other than the sign bit, the rest of bits represent the magnitude of the value. An integer X in N-bit signed magnitude representation is interpreted as,

$$X = (-1)^{\alpha_{N-1}} \sum_{i=0}^{N-2} \alpha_i 2^i \quad \alpha_i \in \{0, 1\}. \tag{3.3}$$

For example, the number $X = -149_{10}$ in signed magnitude format is represented as $\alpha = (1)10010101_2$ using 9 bits (the minimum for this number), where (1) is the sign bit. The range of values that can be represented using N-bits signed magnitude is

$$-2^{N-1} + 1 \leq X \leq 2^{N-1} - 1. \tag{3.4}$$

Compared to unsigned binary, an extra bit for the sign bit is required while being able to represent the same values (and negative values in addition). Zero can be represented in two ways, using either a one or zero sign bit. An 8-bit signed magnitude number can encode the values from -127 to 127. Signed magnitude number representation commonly used in Analog-to-Digital Converter (ADC).

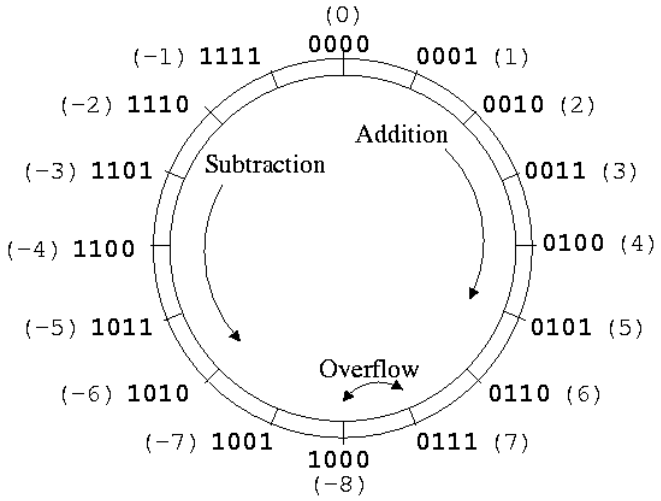


Figure 3.1: Counting wheel for 4-bit two's complement integer numbers [3]

3.1.2.2 Two's Complement Number Representation

Two's complement representation is widely used in computing hardware due to its simplicity and effectives for performing arithmetic operations [1, 25–27]. In two's complement, positive numbers are represented the same way as in unsigned binary and signed magnitude binary. However, negative numbers use represented as the two's complement of their absolute value, where taking the two's complement is equivalent to taking the one's complement (inversion of each bit) and adding one [1, 25, 28].The numbers can be shown in a wheel diagram format in Figure 3.1 [3]. One of useful properties of two's complement addition is that which the final sum of a string of numbers maybe correct, the partial sums could involve overflows [1]. The most significant bit is also called the sign bit and indicates a negative number if the bit is 1. The value of a number in N-bits two's complement can be obtained from [26],

$$X = -\alpha_{N-1}2^{N-1} + \sum_{i=0}^{N-2} \alpha_i 2^i \quad \alpha_i \in \{0, 1\}. \tag{3.5}$$

For instance, the number $X = -149_{10}$ is $\alpha = 101101011$ two's complement with 9 bits. Two's complement with N bits can represent the range [1]:

$$-2^{N-1} \leq X \leq 2^{N-1} - 1. \tag{3.6}$$

From Equation (3.6), when n is 8, the integers from -128 to 127 can be represented using two's complement.

3.2 REDUNDANT NUMBER SYSTEM

A redundant number system is a numeral system that uses more than β digits to represent a radix-β digit [25, 29, 30]. In this way, most of numbers have more than one representation. In general, a redundant number system allows negative values. With a radix of β and N digits, the value of a number X is found by [26],

$$X = \sum_{i=0}^{N-1} \alpha_i \beta^i, \quad \alpha_i \in S, |S| > \beta, \tag{3.7}$$

$$S = \bigcup_{N=0}^{\beta-1} S_N, \tag{3.8}$$

$$S_N = \{\alpha_i : \alpha_i \mod \beta\}, \tag{3.9}$$

$$|S_N| \geq 1. \tag{3.10}$$

For example, when $\beta = 2$,$N = 8$ and $\alpha \in \{-1, 0, 1\}$. Here, symbol $\bar{1}$ is use to represent the value of -1. In this way, the value $X = 115_{10}$ can be represented as either 10010101, $1010\bar{1}\bar{1}01$, $100110\bar{1}\bar{1}$ or $\alpha 4 = 1010\bar{1}0\bar{1}\bar{1}$. The representation of a number in redundant number system is not unique [25].

Following the Equation (3.7), the case where the radix is $\beta = 2$ is a redundant binary representation(RBR). RBR is not other binary number representation, a carry-free addition can be allowed in RBR, but it slow down the bitwise logical operation [30]. The RBR is a signed-digit representation when the digits have signs [29, 30].

However, some number representations share some of the characteristics of RBR but they also have their own constrained conditions to represent values. In some situations these constrains lead to a number representation that is redundant number representation, we call these pseudo-redundant numbers. The representation like Canonical sigend digit (CSD) and sum of power-of-

two(SPT) number representation system are pseudo-redundant number representations. These two number representations are introduced in Section 3.3 Section 3.4.

3.2.1 Hamming Weight and Hamming Distance

Hamming weight refers to the number of non-zero symbols in a number representation string, denoted by K [31–33]. For binary number representation, the Hamming weight is the numbers of 1. For example, the unsigned binary representation of 149 is 10010101 and the Hamming weight is K = 4.

Hamming distance can be interpreted as the number of bits which need to be changed in order to convert one string to another [31–33].

Hamming weight and Hamming distance are commonly used metrics when implementing digital filters in software and hardware. Reducing the Hamming weight and Hamming distance reduces computation, especially with redundant number systems [31].

3.3 CANONICAL SIGNED DIGIT(CSD) NUMBER REPRESENTATION

Canonical Signed Digits(CSD) representation is a number representation for reducing the complexity of coefficients of an FIR filter [27, 34–40]. The way the CSD representation can make coefficients simplicity is because it can represent the coefficient using the minimum hamming weight [27, 34, 36].In this way, the number of multipliers can be decreased significantly [27]. It has been successfully used by others [41, 42].

In CSD representation is given by [26],

$$X = \sum_{i=0}^{N-1} \alpha_i 2^i \quad \alpha_i \in \{-1, 0, 1\} \quad \alpha_i * \alpha_{i+1} = 0, \tag{3.11}$$

with the noted constrain that its adjacent digits cannot be both non-zero digit. Equation (3.11) includes the property that two adjacent digits cannot both be non-zero digits. For example, when $X = 149_{10}$, $N = 8$, $\alpha = 1001010$. In addition, CSD number representation has a unique

representation for each number. The use of CSD representation can reduce arithmetic computation compared to using two's complement [27, 40]. CSD number representation has a 50 % probability of a zero digit, its maximum number of non-zero digits is $n/2$, while the probability is around 2/3 for being a zero-digit.

3.4 SUM OF POWER-OF-TWO(SPT) NUMBER SYSTEM

In binary computation, it is a simple process that multiplying a number by an integer power-of-two. Hence, in order to reduce the multiplications in the binary arithmetic it is good to represent the number using all integer power-of-two terms [6,18,43,44]. In the SPT space, there are three representations with their own constraint. In this section, we give the details of these three representations.

3.4.1 Signed Magnitude Sum of Power-of-Two(SMPT$_K$) Number Representation

SMPT$_K$ is a method that uses the integer power-of-two terms to represent a number [6,18,44,45]. It is also a number system with base 2 and zero(1) or one(1) is its digits but it allocates a "sign bit" in the most significant bit to indicates the number is negative or positive. If the sign bit is on as 1, the number is negative whereas the number is positive when the sign bit is 0. Besides, the K is the Hamming weight of the SMPT$_K$ number representation. A whole number X can be represented to a precision 2^Q by N bits using SMPT$_K$ as [6,18,26],

$$X = (-1)^{\alpha_{N-1}} \sum_{i=0}^{K-1} \alpha_i 2^{q^{(i)}}, \alpha_i \in \{0,1\} \quad , \tag{3.12}$$

where $Q \leq q^{(i)} \leq N-1$. K is the number of the Hamming weight. When the $K = 2$ and $Q = 8$, a number $X = -149_{10}$ can be represented as $\alpha = (-1)10010000$ using $N = 9$ bits (the minimum for this number). The advantage of this representation is that it can limit the Hamming weight of the representation, which can effectively reduced the number of non-zero digits [44].

3.4.2 Signed Digit Sum of Power-of-Two(SPT_K) Number Representation

SPT_K plays a very important role in implement FIR filter designing [44,46–49]. In order to design multiplierless filters, most application using signed power-of-twos term(SPT_K) to represent the coefficient value and signals [6,18]. SPT_K number representation can be interpreted as a method that can represent a number that using the signed integer power-of-two terms also with the number of K hamming weight [18, 42, 44]. It is proposed that representing coefficient in same number of signed power-of-two(SPT) terms is more efficient [44, 45, 47, 50, 51]. Another method is desirable to make all the SPT terms in the least weight length for whole filter [6, 18, 44, 47]. It is also be proved if coefficient value are assigned with different number of SPT terms but keeping the total number of SPT terms fixed had unexpected advantage [6, 18, 44].

In this thesis, we will explore to represent coefficients in the same number of SPT. A number X can be represented as SPT representation with a precision 2^Q by N bits as [6, 18, 26],

$$X = \sum_{i=0}^{K-1} \alpha_i 2^{q^i} \quad \alpha_i \in \{-1, 0, 1\}, \tag{3.13}$$

where $Q \leq q^{(i)} \leq N-1$ and K is the number of SPT terms. For example, when the $Q = 8$, $K = 3$, N is 8 bits, the number $X = 154_d$ can be represented to precise 2^8 as α=10011000.

Actually, it is a good combination to combine limited hamming weight and SPT term together. The SPT_K is more efficient as it requires the minimum hamming weight, so it is more effectively reducing the number of multiplication in the arithmetic [6, 18, 44, 46]. A SPT_K representation for a number is unique sequence [42].

Furthermore, the SPT_K with CSD constraint is widely used in representing the number. The CSD constraint is aiming to make the adjacent two digits in representation cannot be nonzero [6, 18]. In this way, the X can be represented to a precision 2^Q by N bits digit canonic SPT number with K SPT terms as [6, 18]

$$X = \sum_{i=0}^{K-1} \alpha_i 2^{q^i} \quad \alpha_i \in \{-1, 0, 1\}, \quad \alpha_i * \alpha_{i+1} = 0, \tag{3.14}$$

where $Q \leq q^{(i)} \leq N-1$. The constraint $\alpha_i * \alpha_{i+1} = 0$ make sure the adjacent two digits cannot be both non-zero. Also, the maximum of the SPT term is up to K, which means the hamming weight of the this representation is K. In this way, the number $X = 154_d$ can be represented to precise 2^8 as α=10100$\bar{1}$00 when the the $Q = 8$, $K = 3$ and N is 8 bits.

For SPT number representation system, the representable value is related to the K, the greater of value of K, the more numbers can be represented.

3.5 SUMMARY

In this chapter, we introduced the common binary number representation, redundant number representation, CSD number representation and SPT number representation. They all have their own advantages in practice. These number representations are widely used in the digital signal processing field especially for the digital filter design. The redundant number representation can allows addition without using a typical carry in FPGA. CSD number representation can make the essentially multiplierless and the SPT number representation system can specific limit the number of non-zero digits in the computation. In the next chapter, we will implement the FIR LPF filter with different number representations.

Chapter 4

THE COEFFICIENT QUANTIZATION ERROR

The coefficient quantization round-off error has been studies by years, a lot of work is looking into the connection between the round-off noise and filter sensitivity to the coefficient quantization errors [52]. In this chapter, we look into coefficient quantization error and run experiments to implement the FIR LPF using different number representation systems. We provide an analysis and comparison of the distribution of round-off error for quantizing coefficients using the two's complement representation, CSD representation and SPT number representation system. The main purpose of this chapter is to derive the probability density distribution of round-off error of coefficients using SPT number representation systems.

4.1 IMPLEMENTING FIR FILTER METHODS

4.1.1 FIR filter Design Methods

Generally, there are three main popular design methods [2]:

1. Parks-McClellan algorithm: Also known as the Remez Exchange method is the most widely used in implementing FIR filter. It is an iteration algorithm accepts that specification of filter in terms of frequency of stopband, passband, passband ripple, and stopband attenuation. It is popular method because it can optimize all important parameters.

2. Windowing: Windowing methods is a simple and quick technique to design of FIR filter. In order to get desired response, the window is used to shape the impulse response of filters.

3. Direct Calculation: The impulse responses of certain types of FIR filters can be calculated directly from formulas.

4.1.2 FIR LPF Specification

For the purpose of illustration, the following specification is used throughout this chapter for the implementation of FIR low-pass Filter based on the Park-McClellan method:

Approximate infinite filter specification is as below:

1) Filter length, N=32

2) Filter type: low pass

3) Passband edge: 8000 Hz

4) Stopband edge: 12000 Hz

5) Stopband attenuation: 50dB

Table 4.1: Coefficient Table

h(n)	Coefficients
h(0),h(31)	0.003043
h(1),h(30)	0.000184
h(2),h(29)	-0.005620
h(3),h(28)	-0.005110
h(4),h(27)	0.006100
h(5),h(26)	0.012535
h(6),h(25)	-0.002380
h(7),h(24)	-0.022600
h(8),h(23)	-0.011260
h(9),h(22)	0.029008
h(10),h(21)	0.037466
h(11),h(20)	-0.023600
h(12),h(19)	-0.083220
h(13),h(18)	-0.016140
h(14),h(17)	0.193064
h(15),h(16)	0.386975

Table 4.1 shows all the coefficients of the filter, give to six significant digits. The corresponding frequency response of the filter is shown in Figure 4.1.

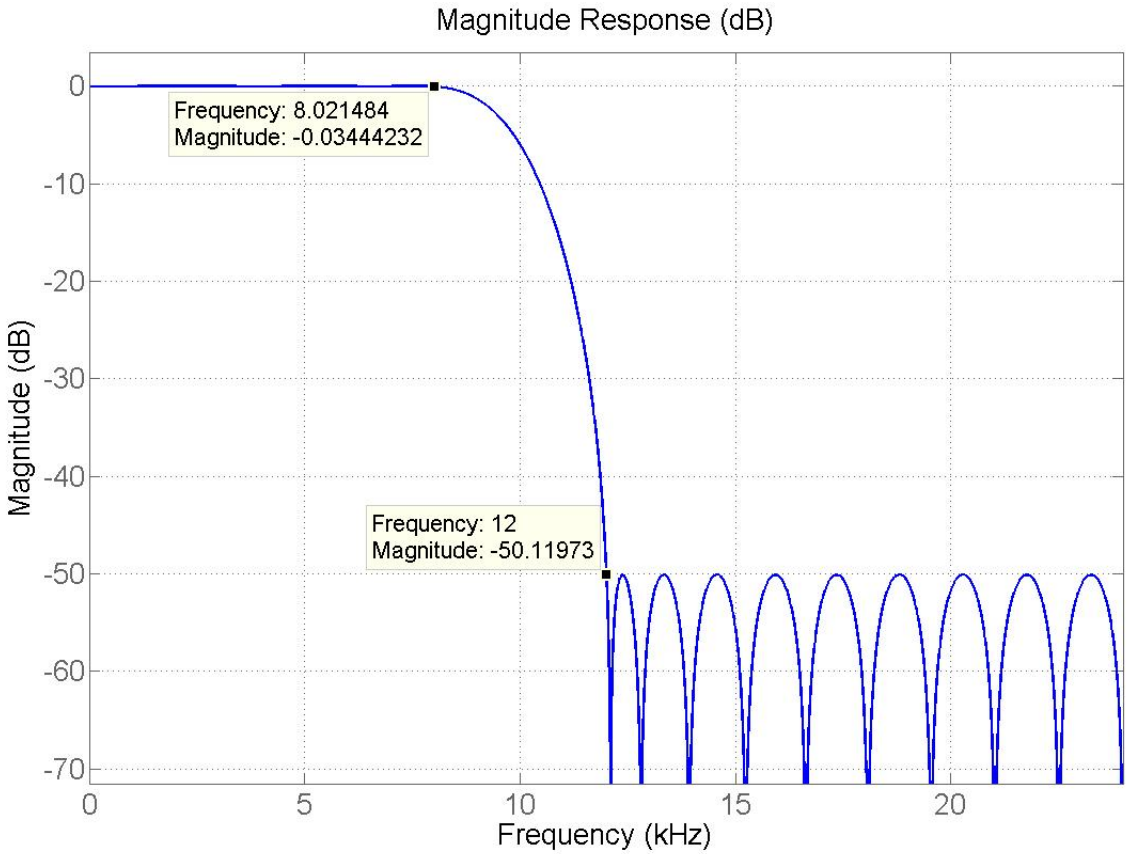


Figure 4.1: Frequency response of approximate infinite filter

4.2 IMPLEMENTING FIR LPF FILTER USING TWO'S COMPLEMENT REPRESENTATION

4.2.1 Quantization of Coefficient of Two's Complement Representation

In general, the result of multiplying two numbers with N bits word-length will be $2N$ bits in length. Due to the limitation of the implemented word-length, truncation or round-off error in the N least significant bits might be caused [1, 2]. As we mentioned in subsection 2.2.5, the quantization round-off error of two's complement is uniform distribution as well as the truncation error. The round-off quantization error of N bits two's complement is [1],

$$-2^{-(N+1)} < E_r < 2^{-(N+1)}. \tag{4.1}$$

The probability density distribution is,

$$P_{Two's}(x) = \begin{cases} \frac{1}{2^{-N}} & |x| \leq \frac{2^{-N}}{2} \\ 0 & \text{otherwise} \end{cases} . \tag{4.2}$$

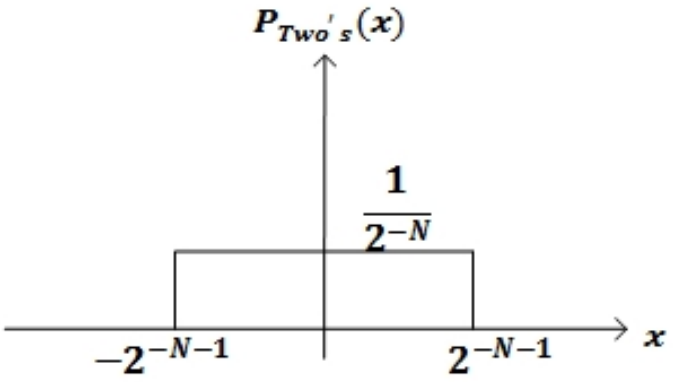


Figure 4.2: Probability density distribution of round-off error for two's complement

Figure 4.2 is the statistical characterization of round-off error for two's complement. For N bits quantization, the error is within $|2^{(-N-1)}|$.

4.2.2 Simulation Results: Implementing an 8 Bits FIR LPF Using Two's Complement Representation

Implementing a FIR LPF using two's complement coefficients are quite common. Table 4.2 shown the coefficients using two's complement format with 8 bits and also give the corresponding representation of two's complement. The round-off quantization error range of 8 bits is [-0.0019531, 0.0019531] with a uniform distribution as shown in Figure 4.3. The multiplication size is 106.

Table 4.2: 8 bits two's complement coefficients and representations

h(n)	Coefficients	Two's quantized coefficients	Two's representation	Multiplication size
h(0),h(31)	0.003043	0.0000	00000000	0
h(1),h(30)	0.000184	0.0000	00000000	0
h(2),h(29)	-0.005620	-0.0078	11111111	7
h(3),h(28)	-0.005110	-0.0078	11111111	7
h(4),h(27)	0.006100	0.0000	00000000	0
h(5),h(26)	0.012535	0.0078	00000001	0
h(6),h(25)	-0.002380	-0.0078	11111111	7
h(7),h(24)	-0.022600	-0.0234	11111101	6
h(8),h(23)	-0.011260	-0.0156	11111110	6
h(9),h(22)	0.029008	0.0234	00000011	1
h(10),h(21)	0.037466	0.0313	00000100	0
h(11),h(20)	-0.023600	-0.0313	11111100	5
h(12),h(19)	-0.083220	-0.0859	11110101	5
h(13),h(18)	-0.016140	-0.0234	11111101	6
h(14),h(17)	0.193064	0.1875	00011000	1
h(15),h(16)	0.386975	0.3828	00110001	2
Total multiplications size				106

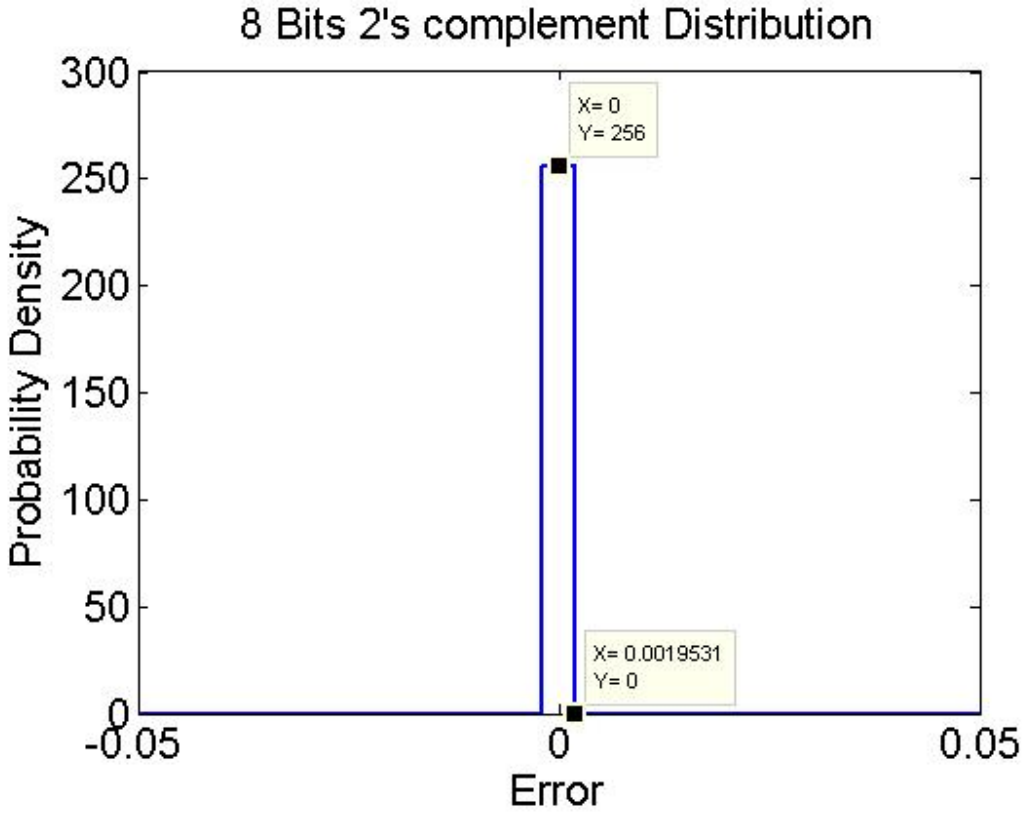


Figure 4.3: Simulation results of 8 bits two's complement representation error distribution

The corresponding frequency response of the filter is given in Figure 4.4. Its passband edge

frequency is nearly the same with desired filter, which is $8000Hz$ and the value of stopband edge frequency is also very close to the desired stopband edge frequency. However, the stopband attenuation is worse, which is -33 dB.

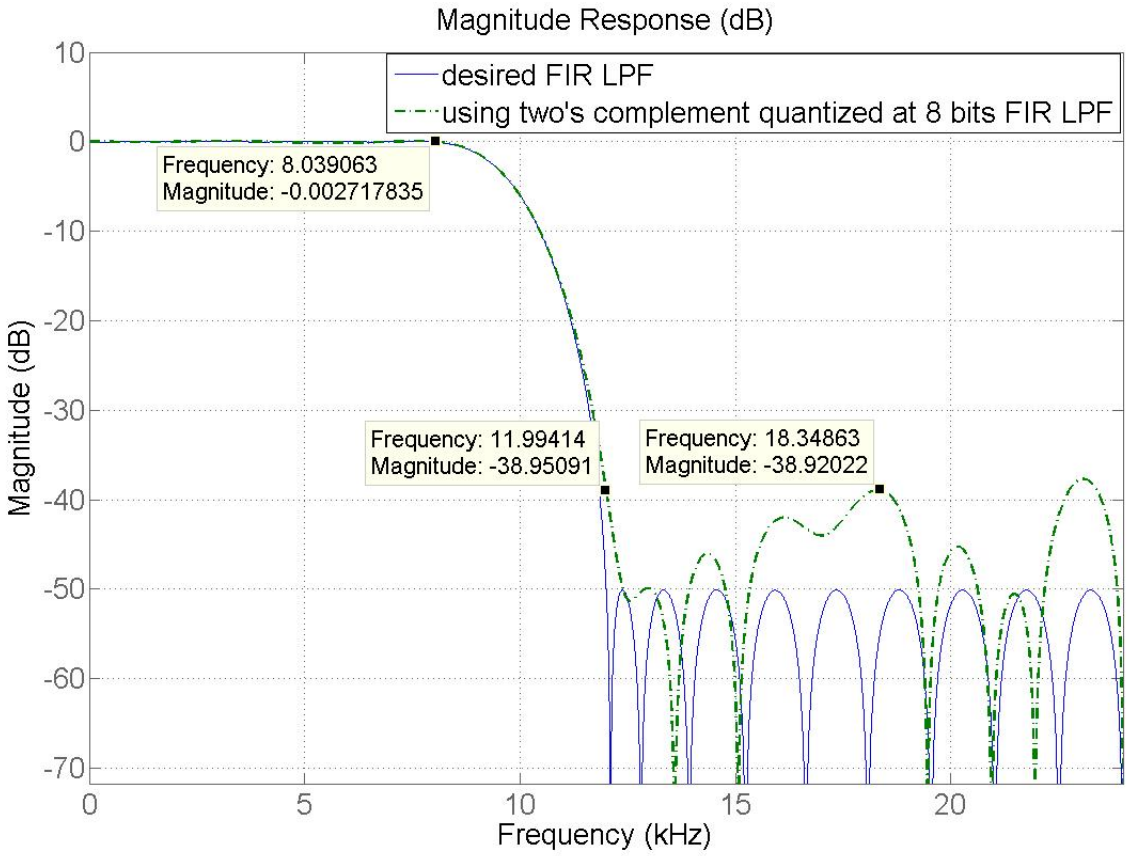


Figure 4.4: The frequency response of two's complement quantized FIR LPF and desired filter

4.3 IMPLEMENTING FIR LPF USING CSD REPRESENTATION

The CSD representation can reduce the number of multipliers by eliminating the nonzero digit in every coefficients value [27] .

Table 4.3: 8 bits CSD representation and CSD coefficients

h(n)	Coefficients	CSD quantized coefficient	CSD representation	Multiplication Size
h(0),h(31)	0.003043	0.0000	00000001	0
h(1),h(30)	0.000184	0.0000	00000000	0
h(2),h(29)	-0.005620	-0.0078	0000000$\bar{1}$	0
h(3),h(28)	-0.005110	-0.0078	0000000$\bar{1}$	0
h(4),h(27)	0.006100	0.0078	00000010	0
h(5),h(26)	0.012535	0.0156	0000010$\bar{1}$	1
h(6),h(25)	-0.002380	0.0000	0000000$\bar{1}$	0
h(7),h(24)	-0.022600	-0.0234	0000$\bar{1}$010	1
h(8),h(23)	-0.011260	-0.0078	00000$\bar{1}$01	1
h(9),h(22)	0.029008	0.0313	0000100$\bar{1}$	1
h(10),h(21)	0.037466	0.0391	00001010	1
h(11),h(20)	-0.023600	-0.0234	0000$\bar{1}$010	1
h(12),h(19)	-0.083220	-0.0859	000$\bar{1}$0$\bar{1}$0$\bar{1}$	2
h(13),h(18)	-0.016140	-0.0156	00000$\bar{1}$00	0
h(14),h(17)	0.193064	0.1953	010$\bar{1}$0001	2
h(15),h(16)	0.386975	0.3906	10$\bar{1}$00101$\bar{1}$	3
Total Multiplication Size				26

4.3.1 Quantization of Coefficient of CSD representation

For CSD coefficient quantization, the round-off error also the uniform distribution as for the two's complement representation. N bits CSD representation, the quantization error will be [6],

$$-2^{-(N+1)} < E_{r_{CSD}} < 2^{-(N+1)}, \tag{4.3}$$

where the quantization step is uniform, the round-off error E_r is also uniform distribution.

4.3.2 Simulation Results: Implementing an 8 Bits FIR LPF Using CSD Representation

The tolerant range of the frequency response of the FIR LPF is influenced by the word-length as well as non-zero digits. The word-length constraints here is $N = 8$ bits. Table 4.3 shows the CSD quantized coefficients and representations. There are at most 4 non-zero digits in each coefficients. According to Table 4.3, the multiplication size is 26.

Figure 4.5 shows the simulation result of Matlab for the 8 bits CSD representation quantization round-off error. It is obviously that the round-off error obeys the uniform distribution, the range of the error is the same with two's complement, which is within $[-0.0019531, 0.0019531]$.

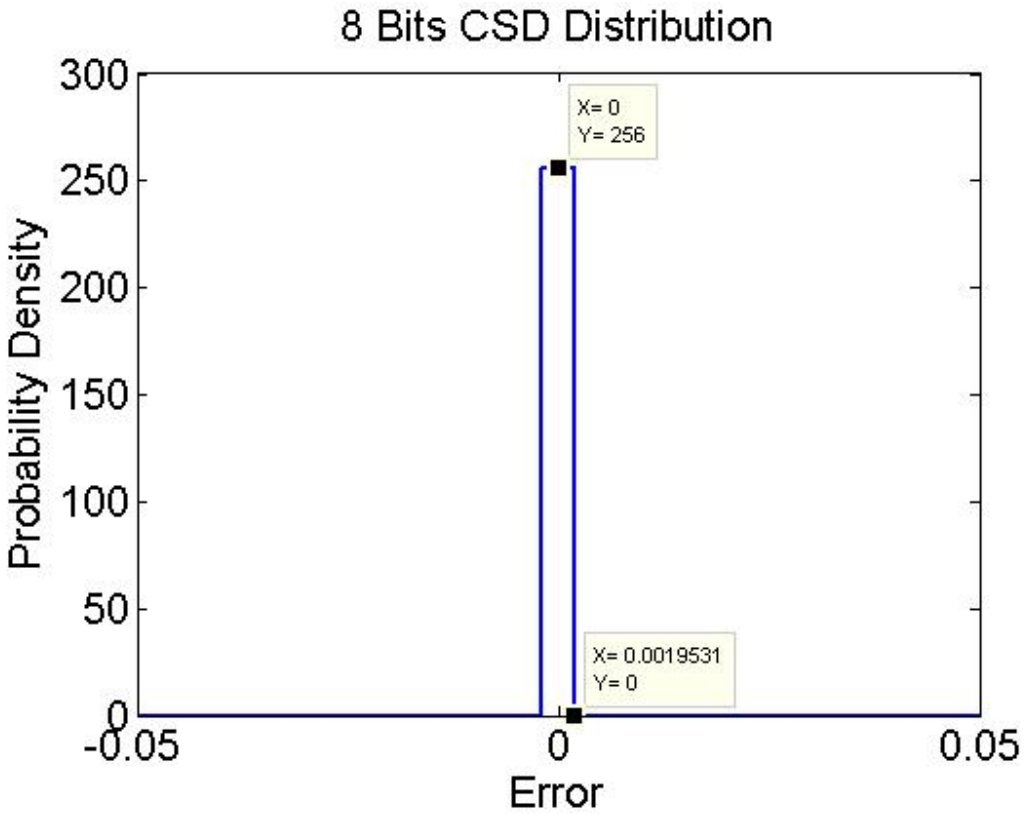


Figure 4.5: Simulation Result of 8 bits CSD representation error distribution

The frequency response of the FIR LPF using CSD coefficients is given in Figure 4.6. Compared with the desired FIR LPF, the stopband attenuation is worse and the stopband frequency is smaller than 12000 Hz whereas passband and passband ripple is as good as the desired filter.

4.4 QUANTITATION OF COEFFICIENT USING SMPT$_K$ REPRESENTATION

SMPT$_K$ representation is a representation using K sum of signed magnitude power-of-two(SMPT) terms, where the hamming weight is K [6,18,44]. In this section, we examine the case of $K = 2$.

4.4.1 Quantization of Coefficient of Using SMPT$_2$ Representation

For SMPT$_2$ terms representation, the quantized process is different with the representation of two's complement and CSD. As it is constrained by the hamming weight of digits, so the quantization step is not uniform [6,18,44–46,51]. As the value of coefficients are all very small

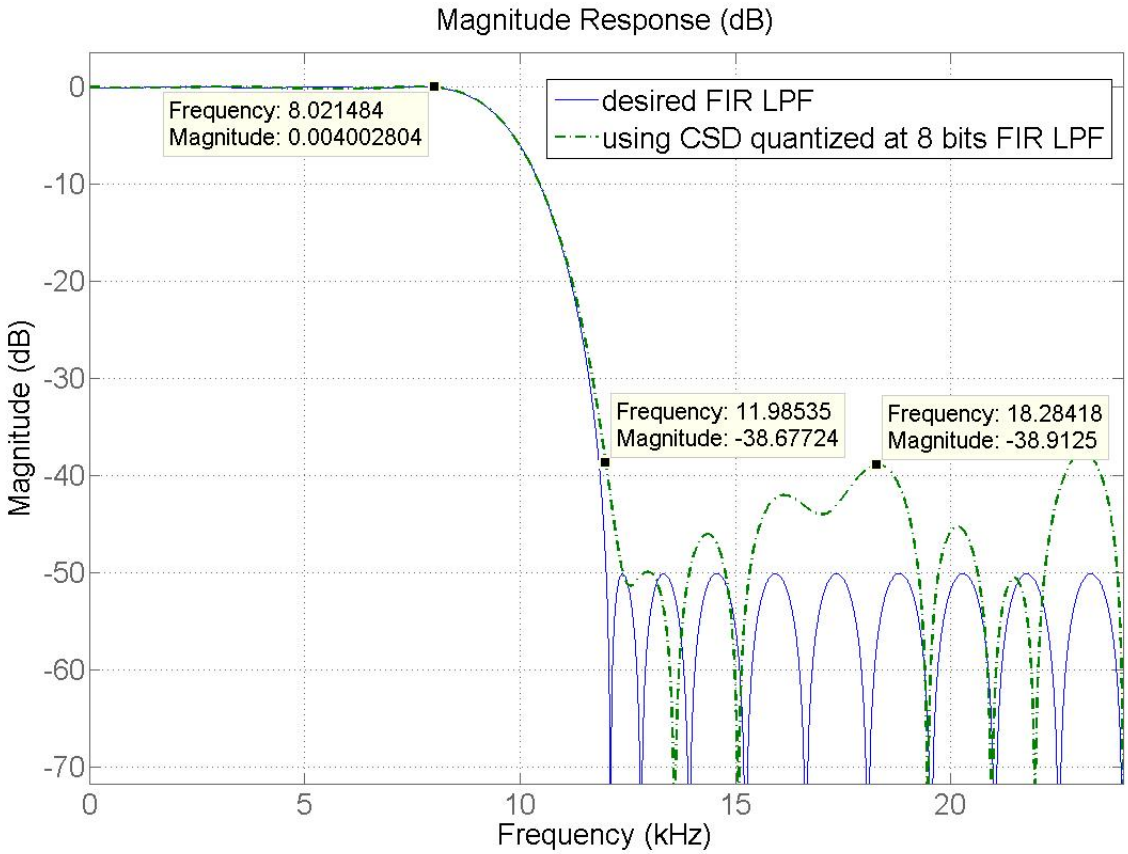


Figure 4.6: Frequency response of CSD coefficient FIR LPF and desired FIR LPF

values in the digital filter, the representable value range discussed here are from $[-0.5, -0.5]$. In this way, the error distribution density distribution is also a little different to the results in [18]. This representable value range $[-0.5, 0.5]$ will also be used later in the sections in this chapter. In order to figure out the SMPT$_2$ quantization, it is necessary to show the trend of pattern of SMPT$_2$ representation for different word-lengths.

Figure 4.7 illustrates the probability density distribution when the word-length varies from 1 bit to 20 bits. The error distribution when the word-length from 1 bit to 3 bits are uniform distribution. However, the error distribution follows a staircase profile when the word-length gradually increases and the value of the peak of probability density increases as well. The bottom error still has a bias of 2^{-4}. The trend of the error distribution of infinite word-length using the SMPT$_2$ coefficients can be derived from Figure 4.7, which is shown in Figure 4.8. One property of SMPT$_2$ representation is that the peak is related to the word-length and it is always

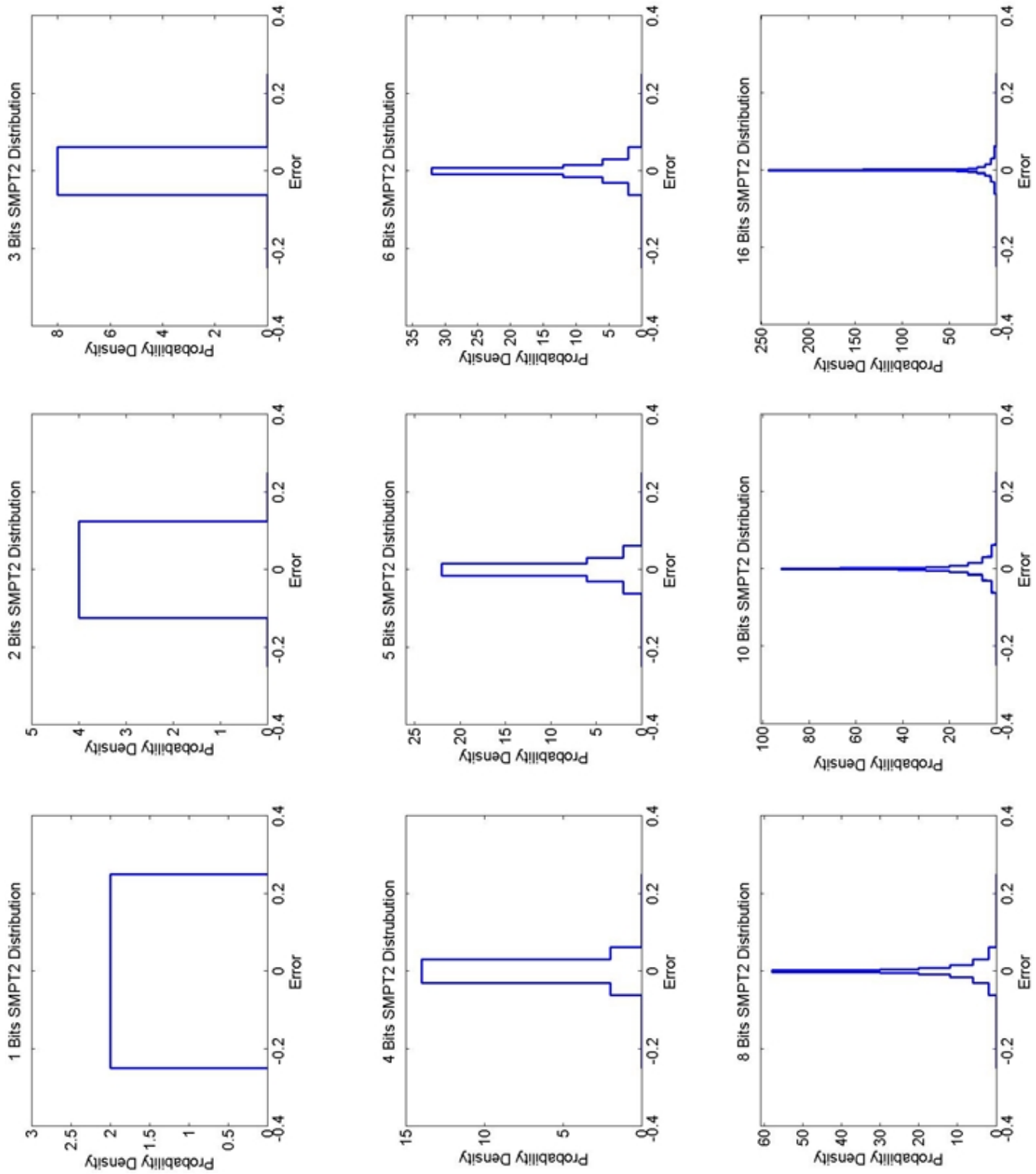


Figure 4.7: SMPT_2 representation error distribution

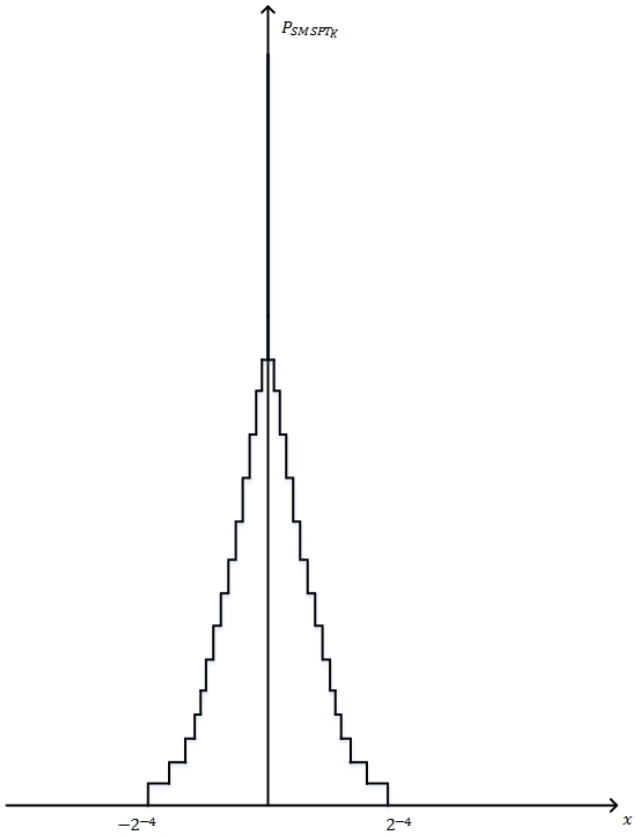


Figure 4.8: The trend approximate line of infinite bits SMPT$_2$ representation error distribution

in the range $|x| \leq 2^{-N-1}$. Another property is the probability remains non-zeros in the interval $[-2^{-4}, 2^{-4}]$, regardless of word-length. Above all, it possible to derive the probability density distribution of signed magnitude with less than 3 bits word-length as,

$$P_{SMPT}(x) = \begin{cases} \frac{1}{2^{-N}} & |x| \leq \frac{2^{-N}}{2} \quad 0 \leq N \leq 3 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{4.4}$$

And when the word-length is greater than 3 bits is,

$$P_{SMPT_2}(x) = \begin{cases} N^2 - N + 2 & |x| \leq \frac{2^{-N}}{2} \\ i^2 + i & \frac{1}{2^{i+4}} \leq x \leq \frac{1}{2^{i+3}} \quad i \in (1, 2, 3..., N-3) \quad N \geq 4 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{4.5}$$

Table 4.4: 7 significant bits $SMPT_2$ coefficients and representation

h(n)	Coefficients	$SMPT_2$ quantized coefficients	$SMPT_2$ representation	Multiplication Size
h(0),h(31)	0.003043	0.0000	(0)0000000	0
h(1),h(30)	0.000184	0.0000	(0)0000000	0
h(2),h(29)	-0.005620	-0.0078	(1)0000001	1
h(3),h(28)	-0.005110	-0.0078	(1)0000001	1
h(4),h(27)	0.006100	0.0078	(0)0000001	0
h(5),h(26)	0.012535	0.0156	(0)0000001	1
h(6),h(25)	-0.002380	-0.0078	(1)0000001	1
h(7),h(24)	-0.022600	-0.0234	(1)0000011	2
h(8),h(23)	-0.011260	-0.0078	(1)0000001	1
h(9),h(22)	0.029008	0.0313	(0)0000100	0
h(10),h(21)	0.037466	0.0391	(0)0000101	1
h(11),h(20)	-0.023600	-0.0234	(1)0000011	2
h(12),h(19)	-0.083220	-0.0781	(1)0000001	1
h(13),h(18)	-0.016140	-0.0156	(1)0000010	1
h(14),h(17)	0.193064	0.1875	(0)0011000	1
h(15),h(16)	0.386975	0.3750	(0)0110000	1
Total Multiplication Size				28

4.4.2 Simulation Results: Implementing an 8 bits FIR LPF Using $SMPT_2$ Representation

$SMPT_2$ is using two SPT terms to represent the coefficients needs an extra bit to show the signed property. Table 4.4 depicts the 8 significant bits $SMPT_2$ quantized coefficients and representation format. According to Table 4.4, the multiplier implement size is 28. However, as the sign bit needs an extra bit, it also costs other resources.

From Equation 4.5, when the $N = 8$, the probability density distribution is,

$$P(x) = \begin{cases} 58 & |x| \le 2^{-9} \\ 30 & 2^{-9} \le |x| \le 2^{-8} \\ 20 & 2^{-8} \le |x| \le 2^{-7} \\ 12 & 2^{-7} \le |x| \le 2^{-6} \\ 6 & 2^{-6} \le |x| \le 2^{-5} \\ 2 & 2^{-5} \le |x| \le 2^{-4} \\ 0 & |x| \le 2^{-4} \end{cases}, \tag{4.6}$$

Figure 4.9 is the Matlab simulation results of the probability density distribution using $SMPT_2$

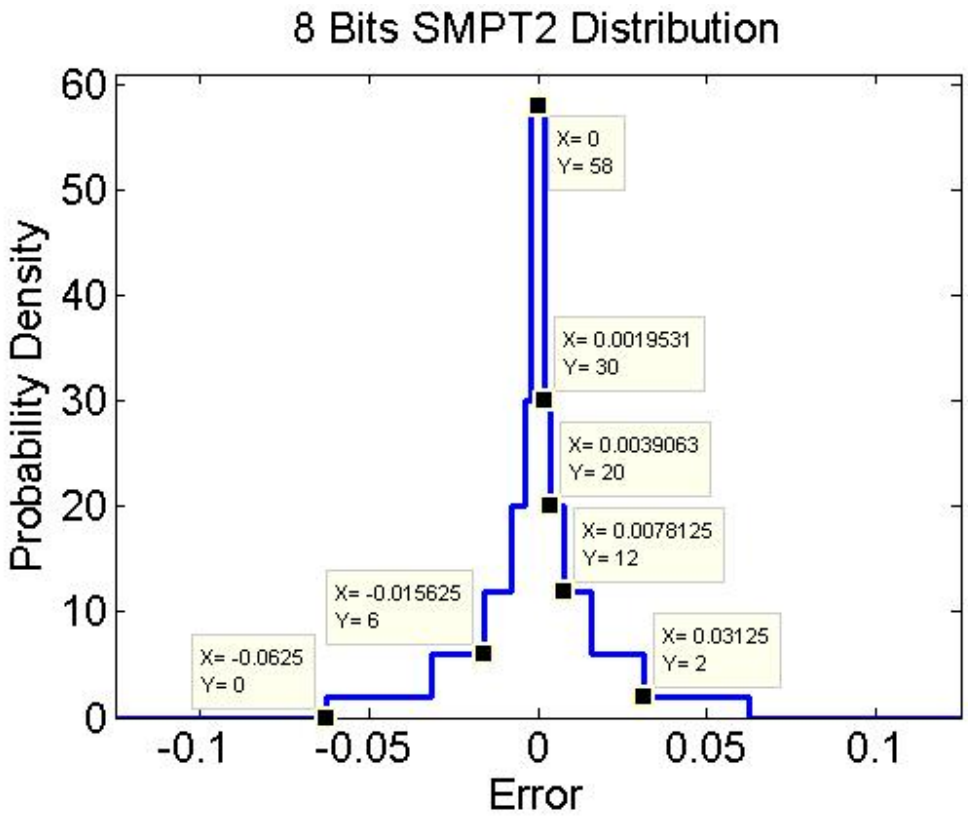


Figure 4.9: 8 bits SMPT$_2$ representation error distribution simulation result

representation when the word-length is 8 bits. It is can be seen that the distribution of round-off error follows a staircase profile. The tag in the staircase profile shown the probability of different error and the results are in agreement with Equation 4.6.

Matlab simulation of frequency response of related filters are shown in Figure 4.10. The dash-dot line is quantized coefficients with 8 bits SMPT$_2$ representation. Its stopband attenuation is worse than desired filter but the passband edge and stopband edge are almost the same.

4.5 QUANTIZATION OF COEFFICIENT USING SPT$_K$ REPRESENTATION

Using SPT$_K$ representation for the coefficients, the digital filter are essentially multiplierless [6, 18, 23, 44, 45, 51, 53]. In this section, the round-off error is introduced when the coefficient are rounded to two of SPT terms.

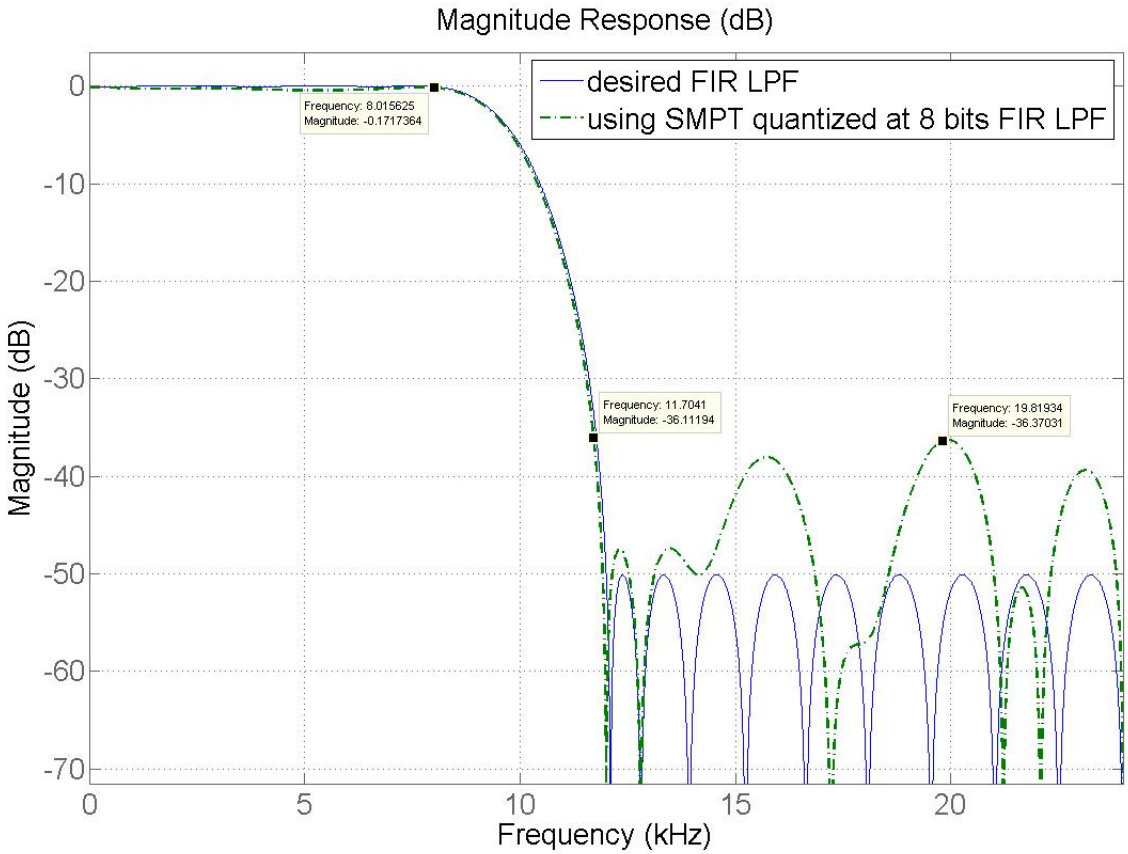


Figure 4.10: The frequency response of SMPT2 quantized coefficients and desired filter

4.5.1 Quantization of Coefficient Using SPT_2 Representation

When the K = 2, the probability density distribution of SPT_K representation is shown as a staircase profile. Figure 4.11 gives the trend of the probability density when the word-length increases from 2 bits to 10 bits. It is clearly that the value of probability density has a sharp increase when the word-length increases. The distribution is uniform distribution when the word-length is smaller than 5 bits. The formula can be derived as,

$$P_{SPT_2}(x) = \begin{cases} \frac{1}{2^{-N}} & |x| \leq \frac{2^{-N}}{2} \\ 0 & \text{otherwise} \end{cases} \qquad 0 \leq N < 5 \quad . \tag{4.7}$$

However, the distribution becomes non-uniform when the word-length is greater than 4 bits, which is shown as the staircase profile 4.11, so when the word-length is greater than 4 bits, the

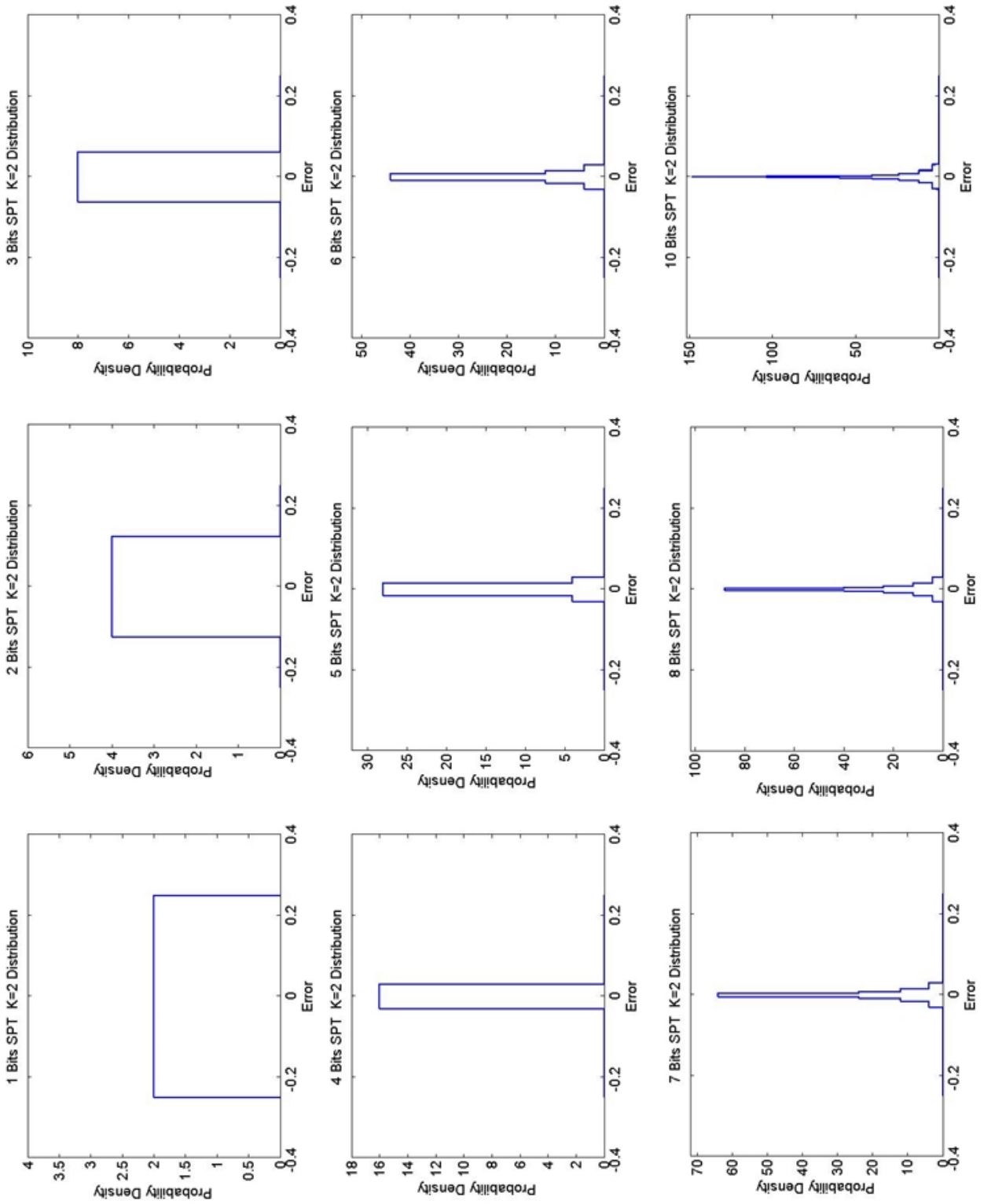


Figure 4.11: SPT$_2$ representation error distribution

distribution of error can be derived as,

$$P_{SPT_2}(x) = \begin{cases} 2(N^2 - 5N + 8) & |x| \leq \frac{2^{-N}}{2} \\ 2(i^2 + i) & \frac{1}{2^{i+5}} \leq x \leq \frac{1}{2^{i+4}} \quad i \in (1, 2, 3..., N-4) \quad N \geq 5 \\ 0 & \text{otherwise} \end{cases} . \quad (4.8)$$

It is worth noting that the peak of the probability density distribution increases with increasing of word-length but the bottom keeps falling in the range of $[-2^{-5}, 2^{-5}]$.

4.5.2 Quantization of Coefficient Using SPT_3 Representation

When the K = 3, the probability density distribution of SPT_3 becomes staircase profile only when the word-length increases to 7 bits. When the word-length is smaller than 7 bits, the probability density distribution of the round-off error is uniform. The trend of the probability density distribution is shown in the Figure 4.12.

4.5.3 Quantization of Coefficient Using SPT_4 Representation

The distribution of the round-off error keeps as uniform when the word-length smaller than the 9 bits. After that the word-length becomes the staircase profile due to the quantization step becomes nonuniform. Figure 4.13 illustrates the trend of probability density distribution of round-off error using SPT_4 representation.

4.5.4 Simulation Results: Implementing an 8 Bits FIR LPF using SPT_K Representation

4.5.4.1 SPK_2 representation

The coefficients are quantized to SPT_2 representation when the word-length is 8 bits in Table 4.5. The multiplication size is 18 using SPT_2 terms representation. The SPT_2 representation can represent both positive and negative numbers due to its extra digit $\bar{1}$.

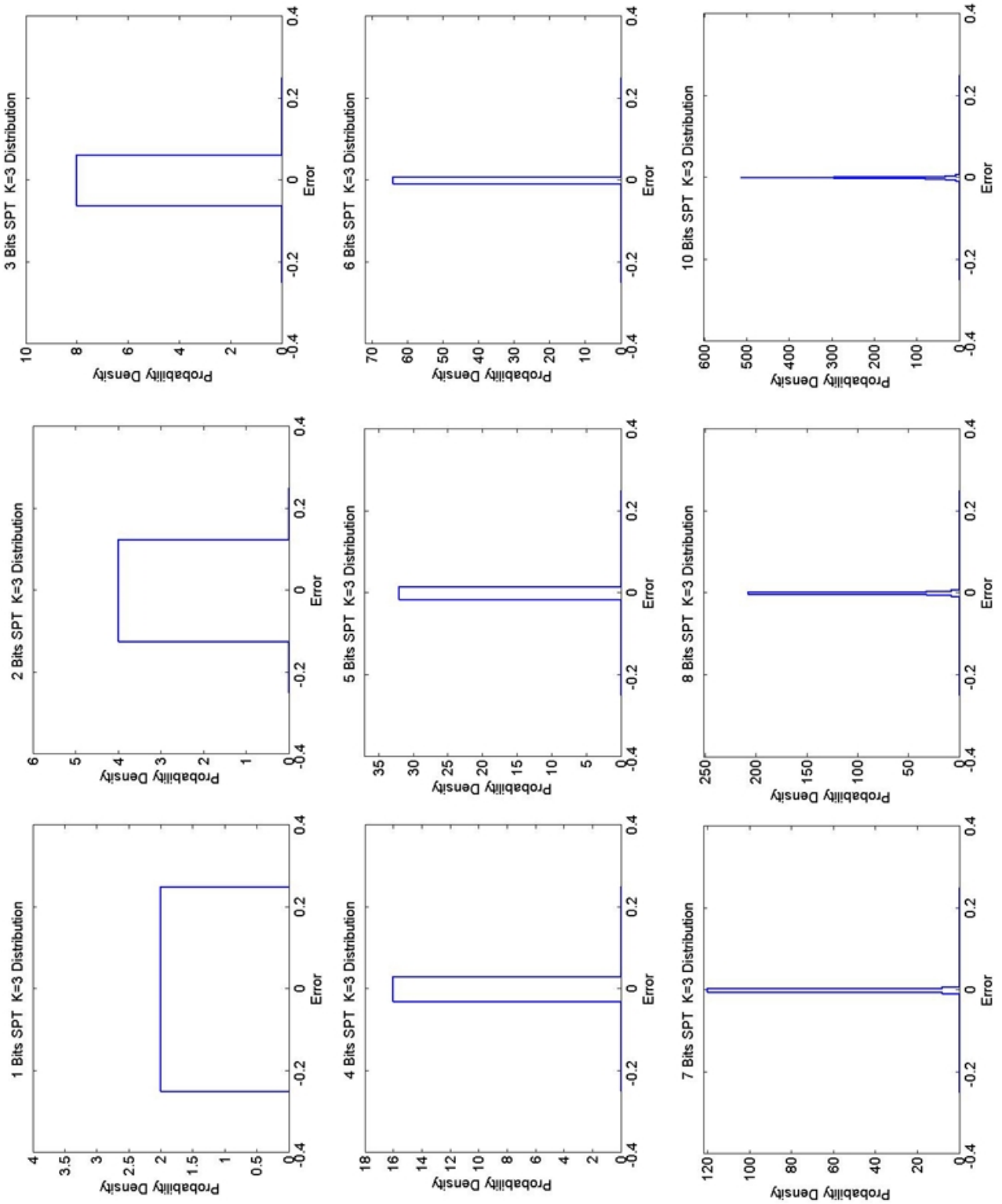


Figure 4.12: SPT$_3$ representation error distribution

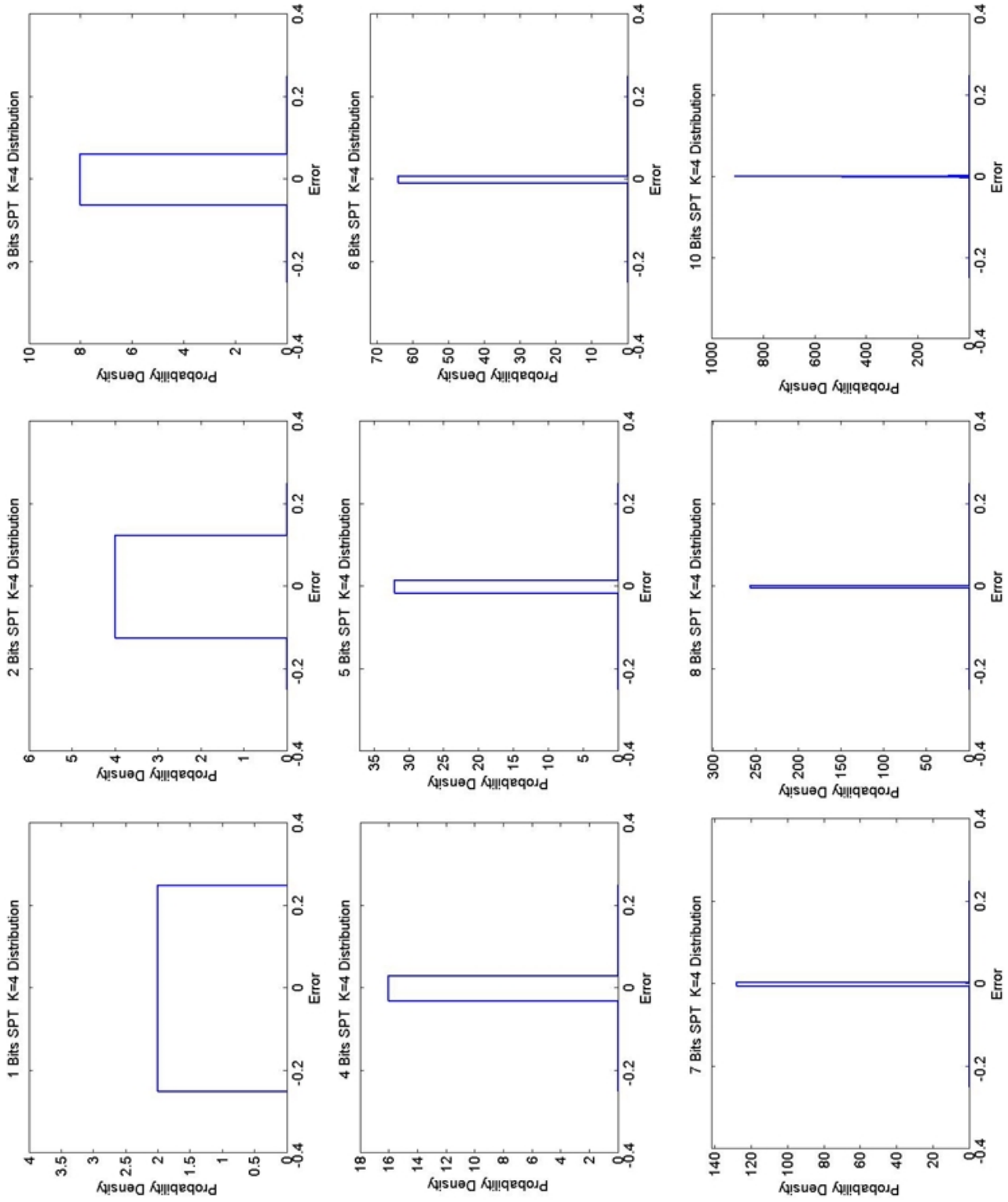


Figure 4.13: SPT_4 representation error distribution

Table 4.5: 8 significant bits SPT_2 coefficients and representation

h(n)	h(n)	SPT_2 quantized coefficients	SPT_2 representation	Multiplication Size
h(0),h(31)	0.003043	0.0039	00000001	0
h(1),h(30)	0.000184	0.0000	00000000	0
h(2),h(29)	-0.005620	-0.0039	0000000$\bar{1}$	0
h(3),h(28)	-0.005110	-0.0039	0000000$\bar{1}$	0
h(4),h(27)	0.006100	0.0078	00000010	0
h(5),h(26)	0.012535	0.0117	00000011	1
h(6),h(25)	-0.002380	-0.0039	0000000$\bar{1}$	0
h(7),h(24)	-0.022600	-0.0234	00000$\bar{1}\bar{1}$0	1
h(8),h(23)	-0.011260	-0.0117	00000$\bar{1}\bar{1}$	1
h(9),h(22)	0.029008	0.0273	0000100$\bar{1}$	1
h(10),h(21)	0.037466	0.0391	00001010	1
h(11),h(20)	-0.023600	-0.0234	00000$\bar{1}\bar{1}$0	1
h(12),h(19)	-0.083220	-0.0781	000$\bar{1}$0$\bar{1}$00	1
h(13),h(18)	-0.016140	-0.0156	00000$\bar{1}$00	0
h(14),h(17)	0.193064	0.1875	00110000	1
h(15),h(16)	0.386975	0.3750	01100000	1
Total Multiplications Size				18

From Equation 4.8, when the $N = 8$ the probability density distribution is,

$$P(x) = \begin{cases} 88 & |x| \leq 2^{-9} \\ 40 & 2^{-9} \leq |x| \leq 2^{-8} \\ 24 & 2^{-8} \leq |x| \leq 2^{-7} \\ 12 & 2^{-7} \leq |x| \leq 2^{-6} \\ 4 & 2^{-6} \leq |x| \leq 2^{-5} \\ 0 & |x| \leq 2^{-5} \end{cases} . \tag{4.9}$$

The maximum round-off error of 8 bits is |0.03125|.

The staircase profile in Figure 4.14 is the simulation results of probability density distribution of a FIR LPF with quantized SPT_2 coefficients when the word-length is 8 bits. The results of probability density shown is in agreement with Equation 4.8.

Figure 4.15 illustrates the frequency response of the experimented FIR LPF. The dash-dot line is the frequency response of quantized SPT_2 coefficients and stopband ripple is about -36 dB, which is worse than the desired FIR LPF. However, the edge of stopband and passband as good

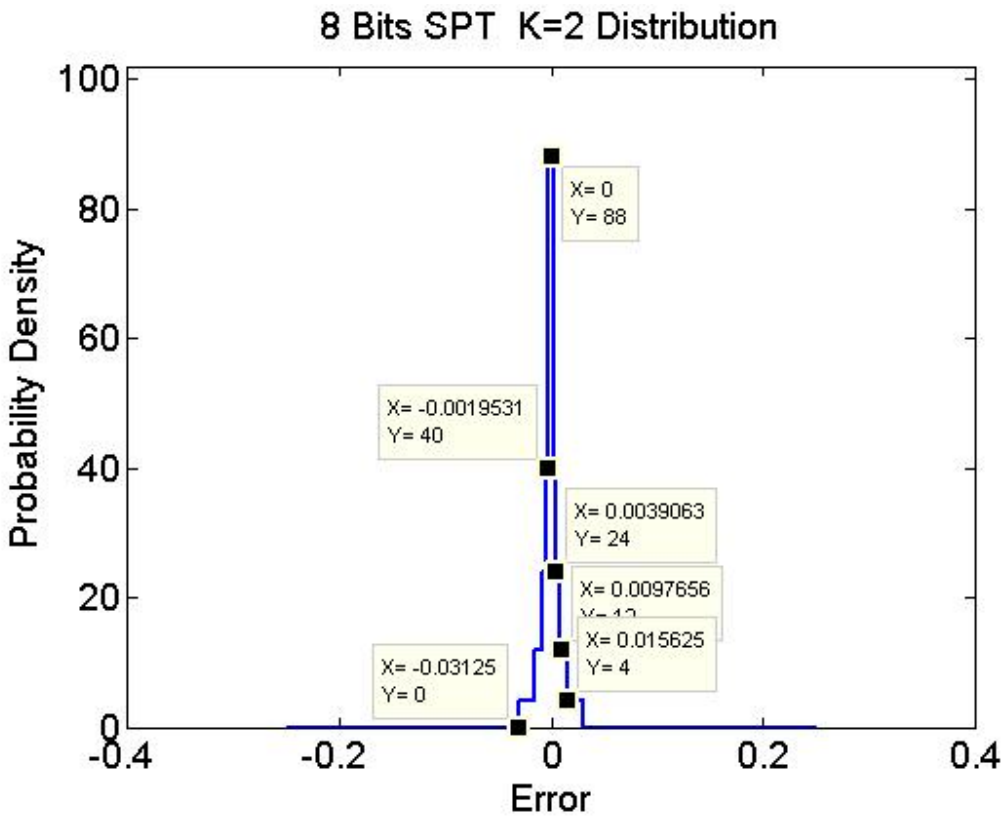


Figure 4.14: 8 bits SPT_2 representation error distribution simulation result

as the desired one.

4.5.4.2 SPK_3 representation

When the $K = 3$, Table 4.6 gives the coefficient using the SPT3 representation. The multiplication size is 26 when using SPK_3 representation, the multiplication size is larger than using the SPT_2 representation.

Figure 4.16 is the frequency response using the SPT_3 representation. Compared with using SPT_2 representation the stopband ripple just improved a little bit, which decreases from -35.71 dB to -36.96 dB.

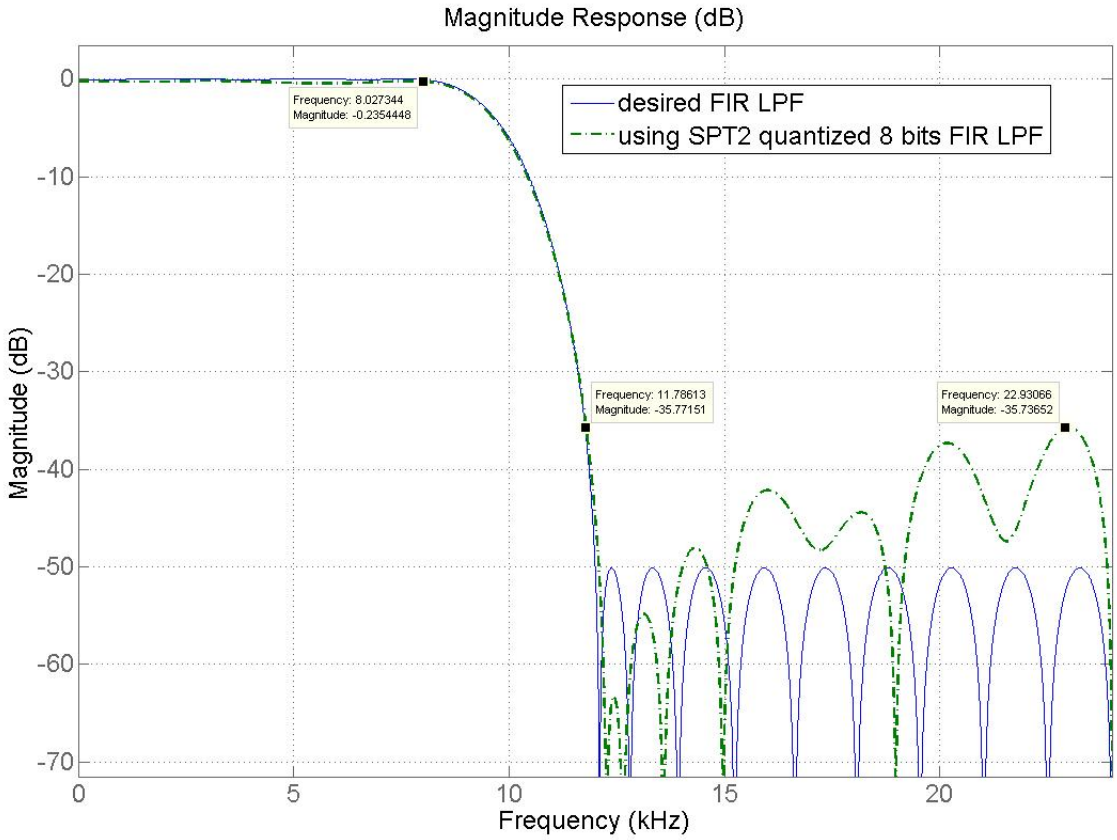


Figure 4.15: The frequency response of FIR LPF using SPT$_2$

Table 4.6: 8 significant bits SPT$_3$ coefficients and representation

h(n)	h(n)	SPT$_3$ quantized coefficients	SPT$_3$ representation	Multiplication Size
h(0),h(31)	0.003906	0.003043	00000001	0
h(1),h(30)	0.000000	0.000184	00000000	0
h(2),h(29	-0.003906	-0.005617	0000000$\bar{1}$	0
h(3),h(28)	-0.003906	-0.005115	0000000$\bar{1}$	0
h(4),h(27	0.007813	0.006100	00000010	0
h(5),h(26)	0.011719	0.012535	00000011	1
h(6),h(25)	-0.003906	-0.002380	0000000$\bar{1}$	0
h(7),h(24)	-0.023438	-0.022600	00000$\bar{1}\bar{1}$0	1
h(8),h(23)	-0.011719	-0.011259	0000000$\bar{1}\bar{1}$	1
h(9),h(22)	0.027344	0.029008	00000111	2
h(10),h(21	0.039063	0.037466	00001010	1
h(11),h(20)	-0.023438	-0.023603	00000$\bar{1}\bar{1}$0	1
h(12),h(19)	-0.082031	-0.083218	000$\bar{1}$0$\bar{1}$0$\bar{1}$	2
h(13),h(18)	-0.015625	-0.016143	00000$\bar{1}$00	0
h(14),h(17)	0.191406	0.193064	00110001	2
h(15),h(16)	0.390625	0.386975	01100100	2
Total Multiplications Size				26

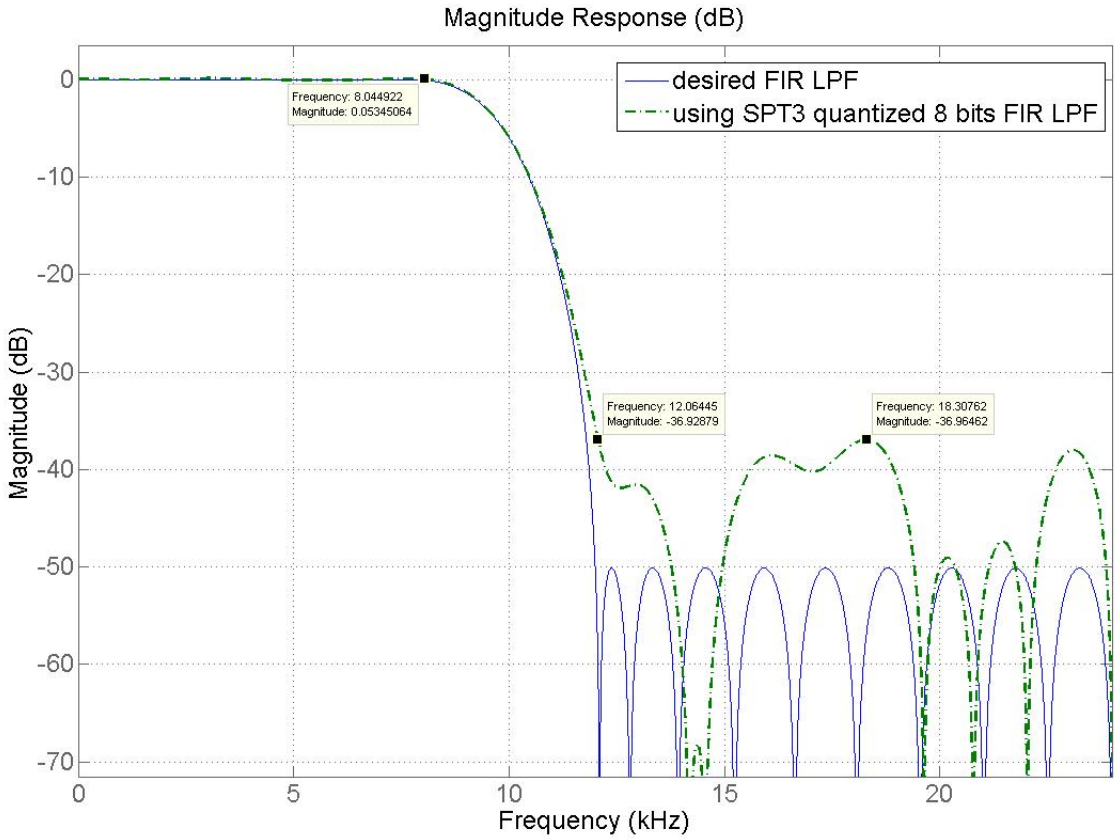


Figure 4.16: The frequency response of FIR LPF using SPT_3 representation

4.5.4.3 SPK_4 representation

When the K increases to 4, but the word-length still remains as 8 bits, so the round-off error cannot be decreased significantly. Table 4.7 gives the SPT_4 representation of coefficients. Compared with the using SPT_3 representation, the multiplication size is just slight larger than using SPT_3 representation, which is 28.

In Figure 4.17, the stopband ripple is -38.97 dB, which decreases around 2 dB than using SPT_3 representation.

Table 4.7: 8 significant bits SPT$_4$ coefficients and representation

h(n)	h(n)	SPT$_4$ quantized coefficients	SPT$_4$ representation	Multiplication Size
h(0),h(31)	0.003906	0.003906	00000001	0
h(1),h(30)	0.000000	0	00000000	0
h(2),h(29	-0.003906	-0.00391	0000000$\bar{1}$	0
h(3),h(28)	-0.003906	-0.00391	0000000$\bar{1}$	0
h(4),h(27	0.007813	0.007813	00000010	0
h(5),h(26)	0.011719	0.011719	00000011	1
h(6),h(25)	-0.003906	-0.00391	0000000$\bar{1}$	0
h(7),h(24)	-0.023438	-0.02344	00000$\bar{1}\bar{1}$0	1
h(8),h(23)	-0.011719	-0.01172	00000$\bar{1}\bar{1}$	1
h(9),h(22)	0.027344	0.027344	00000111	2
h(10),h(21	0.039063	0.039063	00001010	1
h(11),h(20)	-0.023438	-0.02344	00000$\bar{1}\bar{1}$0	1
h(12),h(19)	-0.082031	-0.08203	000$\bar{1}$0$\bar{1}$0(1)	2
h(13),h(18)	-0.015625	-0.01563	00000$\bar{1}$00	0
h(14),h(17)	0.191406	0.191406	00110001	2
h(15),h(16)	0.390625	0.386719	01100011	3
Total Multiplications Size				28

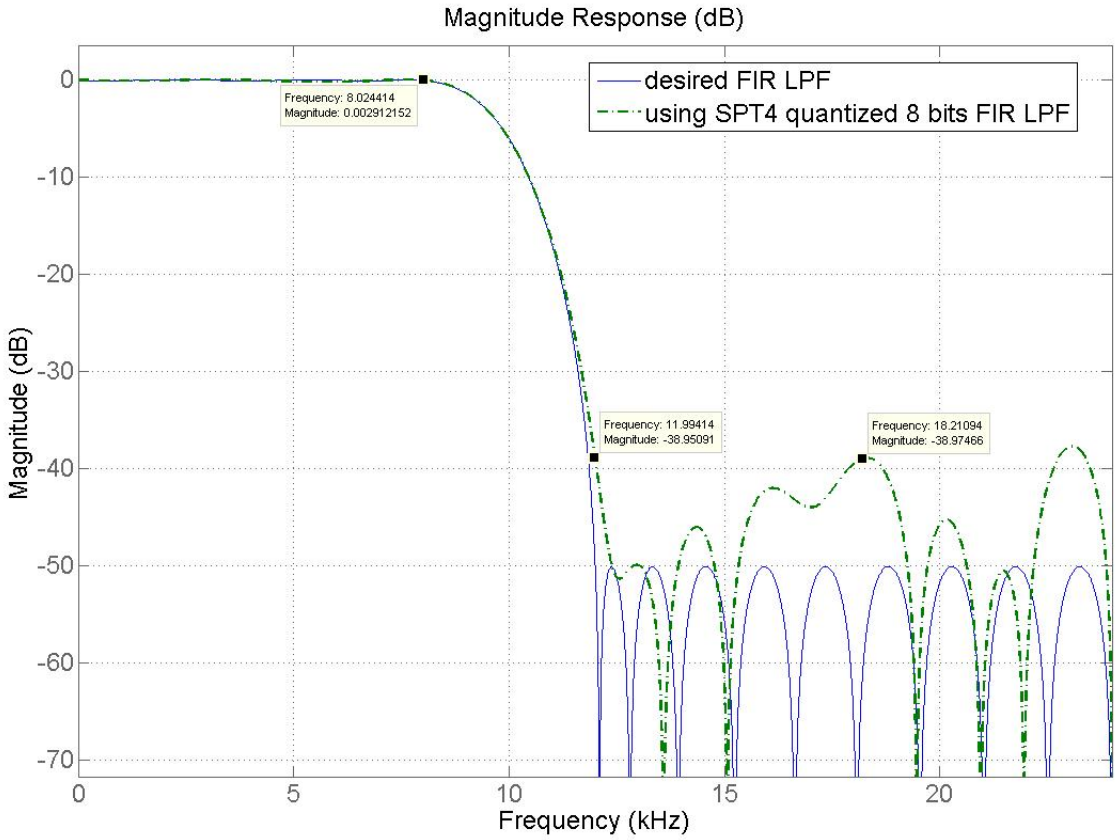


Figure 4.17: The frequency response of FIR LPF using SPT$_4$ representation

4.6 QUANTITATION OF COEFFICIENT USING SPT$_K$ REPRESENTATION WITH CSD CONSTRAINT

In Chapter 3, Section 3.4, the SPT$_K$ with CSD constraint K was already introduced. After the experiments, we found that the trend of round-off error of probability density distribution for representation with CSD constraint of SPT$_2$, SPT$_3$ and SPT$_4$ are the same with the graphs in Figure 4.11, Figure 4.12 and Figure 4.13. When the value of K is smaller than the half of the word-length, the probability density of the error distribution is the same with the SPT$_K$ representation [6, 18]. In this way, when the K = 2, the probability density formula of round-off error using SPT$_2$ representation with CSD constraint is the same result with in the Equation 4.7 and Equation 4.8

4.6.1 Simulation Result: Implementing an 8 Bits FIR LPF using SPT$_K$ Representation with CSD constraint

Figure 4.18 give the simulation results of using the SPT$_2$ representation with CSD constraint. The probability density in Figure 4.18 is the same with the Figure 4.14, which is in the agreement with the Equation 4.8 when the $N = 8$ bits.

It is can be seen from the above simulation result that in the representable value range [-0.5 0.5], when the value of K is smaller than the half of the given word-length, the formula of probability density of round-off error is same with the SPT$_K$ representation.

4.7 ANALYSIS AND COMPARISON

The Figure 4.19 shows that the value of mean error for both two's complement quantized and CSD quantized can be continuously decreased with increasing of the word-length. The reason is the probability density distribution are uniform for both two's complement representation and CSD representation [1, 6, 18]. The round-off error of both are related to the word-length. With increasing word-length, the round-off error can be reduced. The round-off error is $|2^{-N-2}|$.

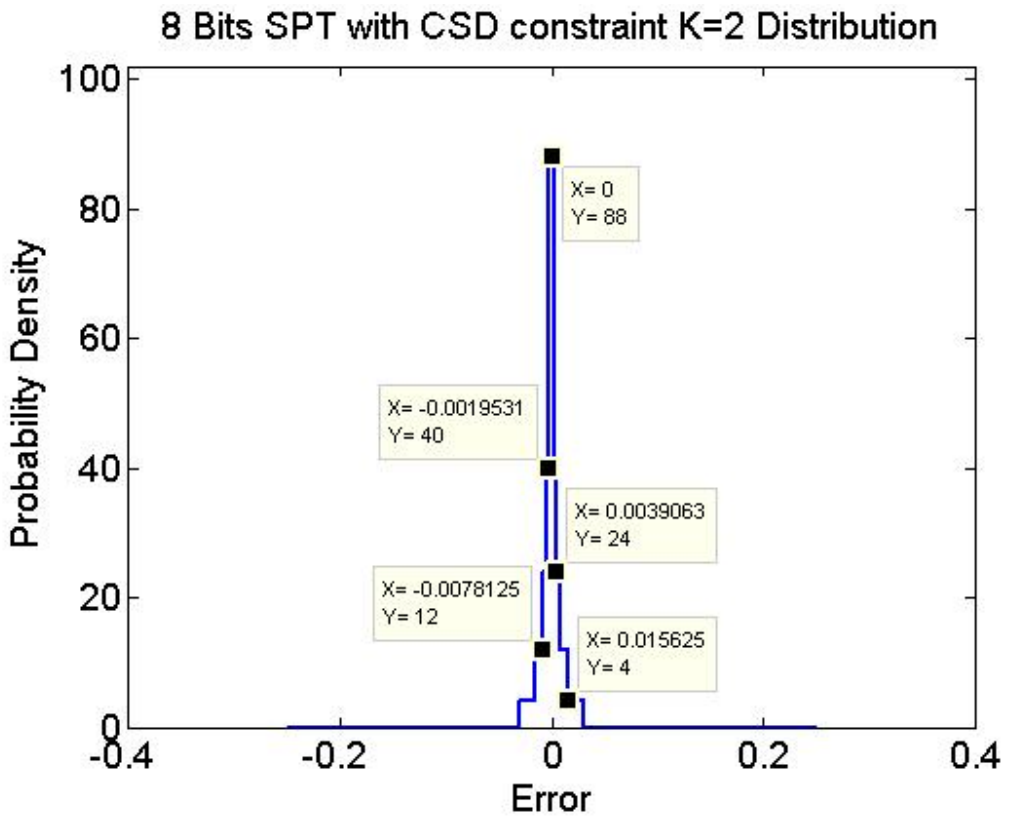
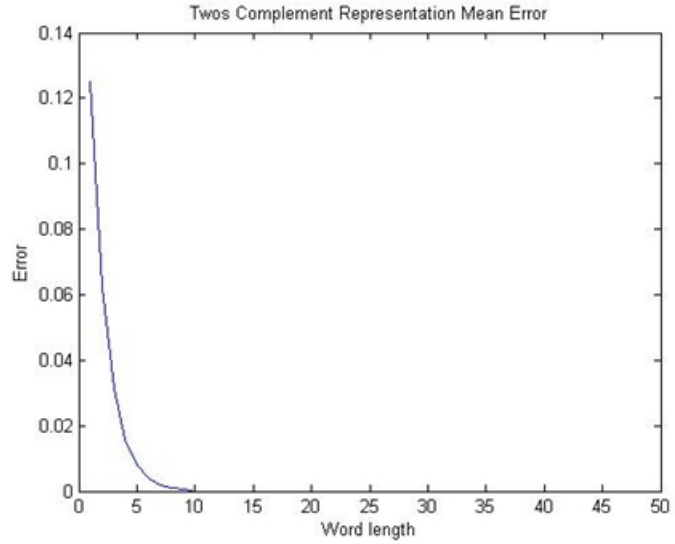


Figure 4.18: Simulation result of 8 bits SPT representation with CSD constraint K = 2 error distribution

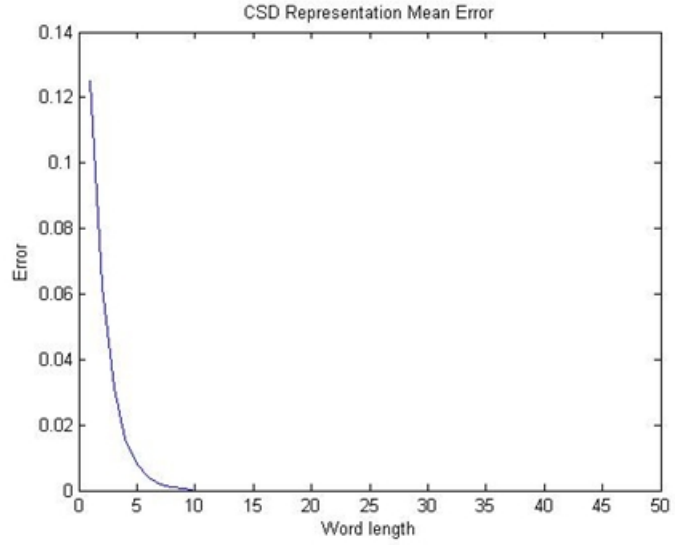
Figure 4.20 gives the frequency response of two's complement and CSD representation when the word-length is 8 bits. It is obviously that the frequency response of them are superposition. It is because their coefficient quantization effects on the magnitude are the same for both two's complement and CSD representation.

Figure 4.21 gives the trend of the mean error for representation of SMPT_2 and SPT_2. Graph (a) illustrates the mean error for SMPT_2 representation. It describes that with the a gradual increase of world length, the mean error has a sharp decrease when the word-length is less than 4 bits but remain the same value after 4 bits, which is agreement with the Equation 4.4 and Equation 4.5 that the word-length smaller than 4 bits is uniform and greater 4 bits became non-uniform.

The same trend for SPT_2 representation in graph (b), the mean error has a big drop when the word-length increase to 5 bits but becomes a constant value, which remains at 0.01388 after 5



(a) The trend of mean error for two's complement



(b) The trend of mean error for CSD representation

Figure 4.19: The trend of round-off mean error for two's complement representation and CSD representation

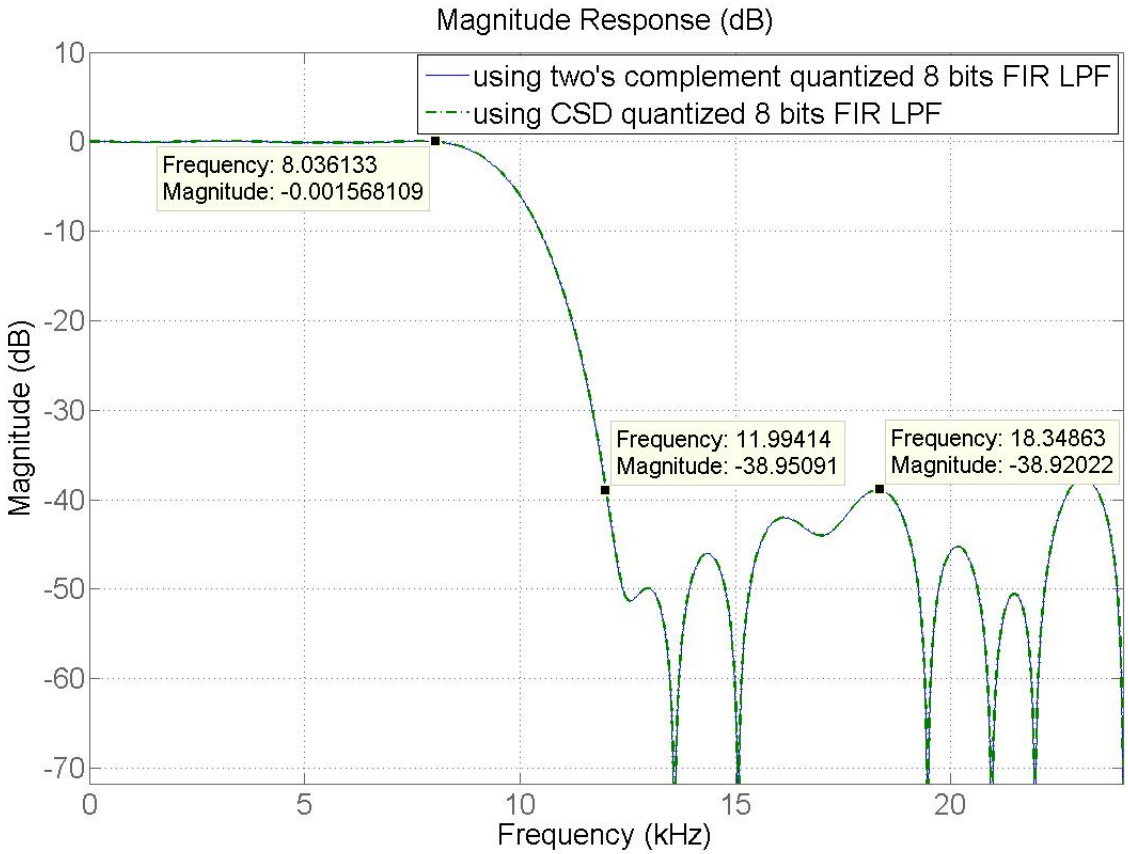
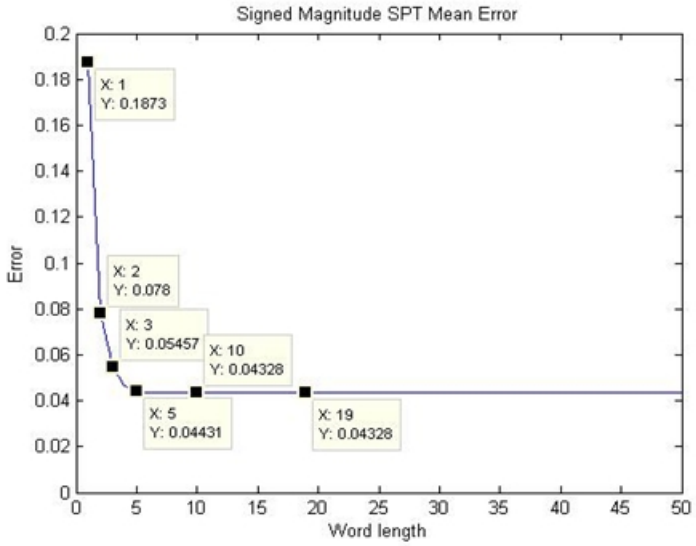


Figure 4.20: The comparison of the frequency response for two's complement representation and CSD representation (8 bits)

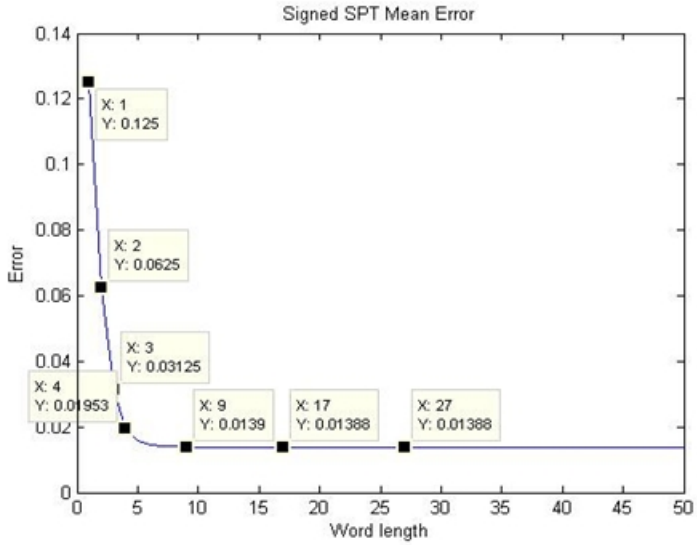
bits. The results is the round-off error for SPT_2 representation cannot be decreased by increasing the word-length due to their quantization characteristics as well.

Although the trend of mean round-off error for both representation are the same but the scales of value are differ. The value of maximum mean error up to 0.1873 for the SMPT_2 whereas for SPT_2 is just 0.0125. It is worth noting that after 5 bits, the steady mean error for SMPT_2 quantization is 0.04328, which is three times the value for the SPT_2 representation (0.0139). It is because the SPT_2 has the extra radix -1, which quantizes the value more precisely.

For SPT_K representation, the value of K and the word-length influence the error of SPT_K representation together. When the word-length N is larger enough to represent a full precise number and the value of K is as possible as close to the word-length, the error can be decreased with increase the K and N together. In particular situation, when the K is equal to the word-



(a) The trend of mean error for signed magnitude SPT2 representation



(b)The trend of mean error for signed SPT2 representation

Figure 4.21: The mean error trend of increasing word-length

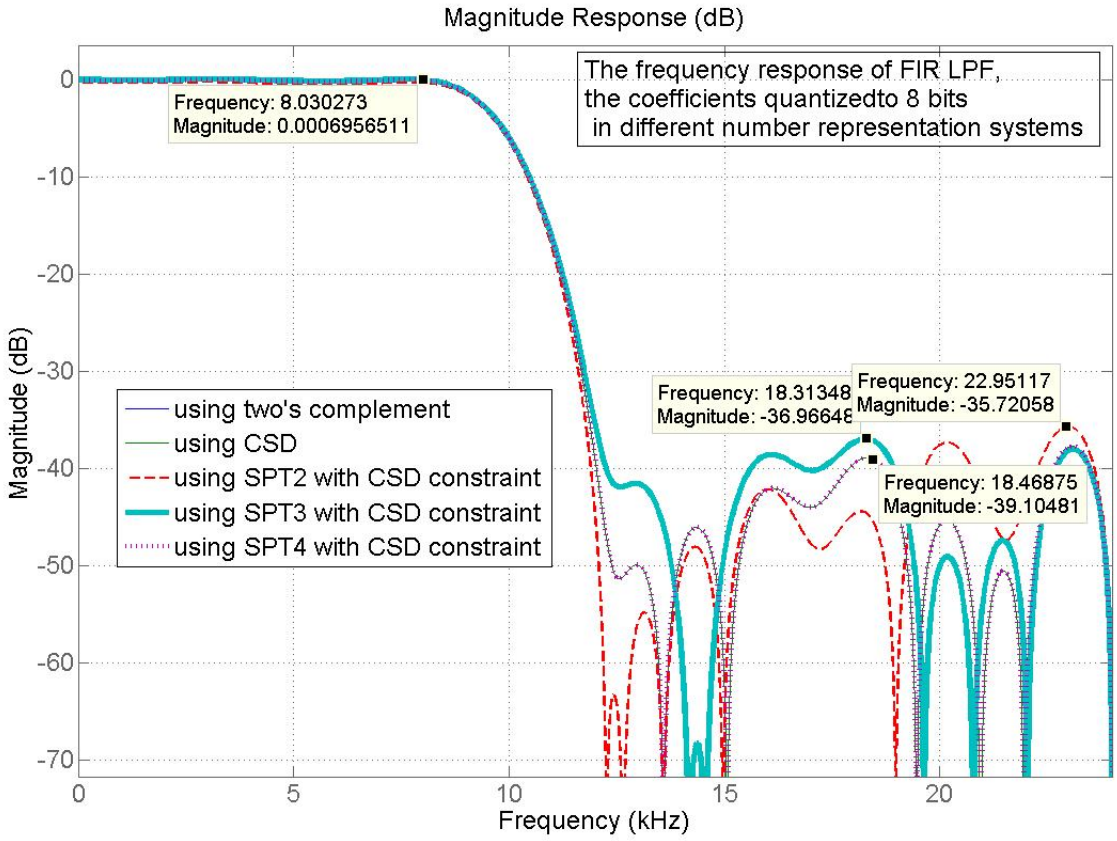


Figure 4.22: The frequency response of using representation of two's, CSD, SPTCSD_2, SPTCSD_3 and SPTCSD_4 at 8 bits

length N, SPT_K is the signed binary representation. The probability density distribution is uniform distribution when the K = N.

For SPT_K with CSD constraint, the probability density of the distribution will more and more closely represent the CSD form when increasing the value of K. When the K equal or greater than the $N/2$, it is becomes the CSD representation, so its round-off error distribution will be uniform [6, 18].

Figure 4.22 gives the frequency response of 8 bits FIR LPF using the two's complement representation, CSD representation and SPT_K with CSD constraint representation. It is obviously can be seen that the frequency response of using CSD representation and SPT_4 with CSD constraint representation are superposition. It is indicates that the round-off errors for this FIR LPF are the same by using both of representations. This result agrees with the statement that when the

Table 4.8: The multiplication size for different numerical system when the word-length is 8 bits

Representation	Multiplication Size
Two's complement	106
CSD	26
$SMPT_2$	28
SPT_2	18
SPT_3	26
SPT_4	28
SPT2 with CSD constraint	18
SPT3 with CSD constraint	24
SPT4 with CSD constraint	26

$K = N/2$, the SPT_K with CSD constraint becomes the CSD representation. It is also shown that with increase of K, the noise floor for using SPT_K with CSD constraint can have a small decrease.

Figure 4.23 is the frequency response of FIR LPF using different number representations when the word-length increase to 12 bits. Compare with the Figure 4.22, the stopband ripple peak for coefficients using SPT_2 representation is still -36.96 dB which is the same when the word-length is 8 bits. The stopband worst attenuation performance for SPT_3 has an improve. For using SPT_4, the worst attenuation decrease from -39.10 dB to -48.12 dB when the word-length is increasing from 8 bits to 12 bits. With increasing of world-length, the stopband performance improved from -39.10 dB to -50.86 dB by using CSD representation. Besides, the frequency response of FIR LPF using SPK_4 representation and CSD representation are not same anymore with increasing the word-length. The reason is K is smaller than the $N/2$, so the probability density of round-off error for SPT_K is non-uniform again when the $N = 12$. From Figure 4.23, it is clear that implementing FIR LPF using CSD presentation and SPT_4 representation is both good choice to achieve desired filter when increasing the word-length to 12 bits.

The Table 4.8 gives the multiplication size when the word-length is 8 bits. When the word-length is 8 bits, the frequency response of two's complement quantized and CSD quantized FIR LPF are the same but the computation using CSD representation is 3 times smaller than using two's complement representation. The multiplication size for $SMPT_2$ representation for $SMPT_2$ is 28, but it will cost more resource due to the $SMPT_2$ needing an extra bit for the signed bit. The SPT_2 representation is the most cost saving one regardless of the larger round-off error. When

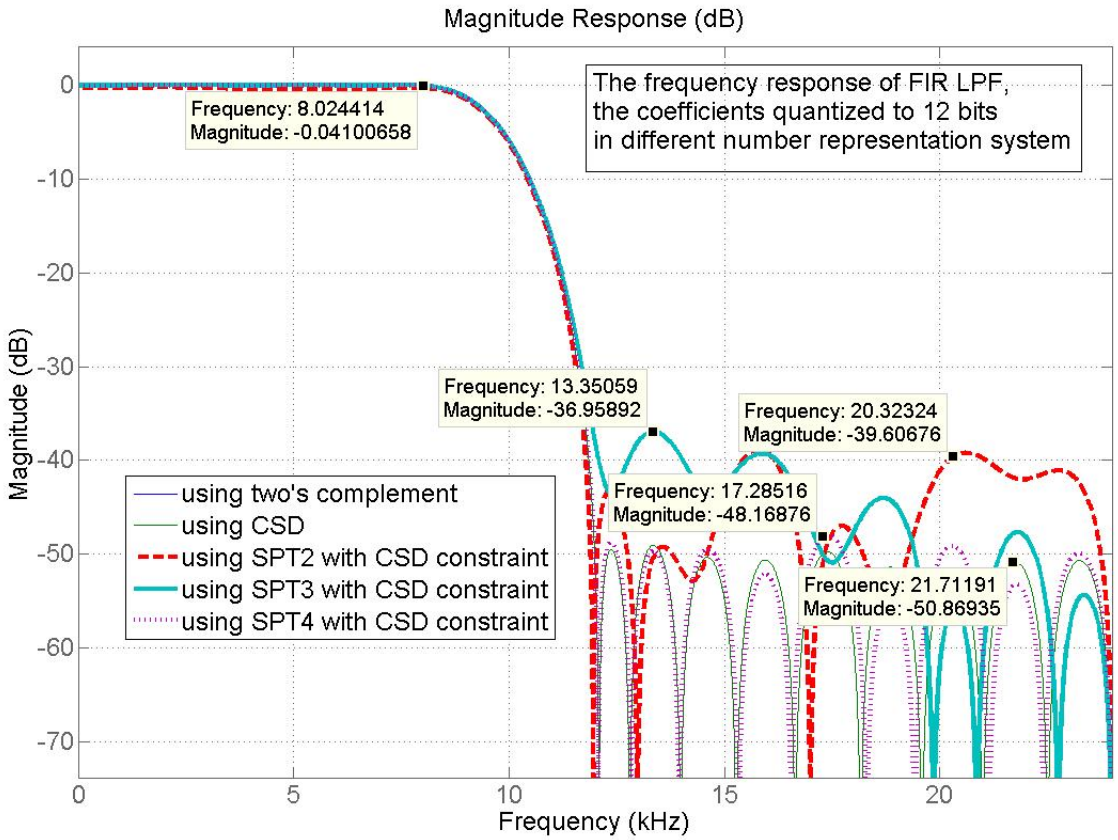


Figure 4.23: The frequency response of using representation of two's, CSD, SPTCSD$_2$, SPTCSD$_3$ and SPTCSD$_4$ at 12 bits

Table 4.9: The multiplication size for different numerical system when the word-length is 12 bits

Representation	Multiplication Size
Two's complement	320
CSD	72
SMPT$_2$	30
SPT$_2$	30
SPT$_3$	54
SPT$_4$	70
SPT2 with CSD constraint	30
SPT3 with CSD constraint	54
SPT4 with CSD constraint	66

the word-length is 8 bits, the multiplication size for CSD representation and SPT_4 representation are the same .

The Table 4.9 illustrates the multiplication size when the word-length goes up to the 12 bits. It is obviously that two's complement still needs much more multiplication size than CSD representation. However, the multiplication size for using SPT_4 representation is smaller than CSD representation. Combining the results got from Figure 4.23, when the word-length increases to the 12 bits, the stopband performance of using SPT_4 representation is nearly the same with using CSD representation, but it requires less multiplication size than CSD representation. In this way, it is more effectively using SPT_4 with CSD constraint representation to implement the desired filter in subsection 4.1.2.

Overall, stopband attenuation is influenced by word-length when using two's complement and CSD representation. For SPT number systems, the stopband worst attenuation is influenced both by the word-length and the value of K. With increasing of the word-length and value of K, the stopband worst attenuation performs better. However, multiplication size also can be reduced by controlling the value of K. Moreover, the best way to implement the FIR LPF specification in subsection 4.1.2 is using SPT_4 with CSD representation.

4.8 SUMMARY

In this chapter, we look into the coefficient quantization of each number representations and implement the FIR LPF using those number representations. The distribution of round-off error for using both two's complement representation and CSD representation are uniform distributed. The formula of distribution of representation of $SMPT_2$ and SPT_2 was derived and the distribution for both of them are all staircase profile, which is caused by the limitation of the using of sum power of two terms.

The results illustrate that the frequency response of using two's complement representation and CSD representation are the same. The mean error trend and value are all the same as well. Their mean error can be decreased to 0 with a gradual increase of the word-length to infinite. However, the mean error for using representation of $SMPT_2$ and SPT_2 can not be decreased

after 5 bits, regardless of the changing of the word-length.

When the value of K increases, the round-off error of SPT_K with K constraint will be more and more close to the CSD representation. When the K increased to the $N/2$, it becomes CSD representation.

Implementing a 8 bits FIR LPF with the same frequency response using CSD representation can consume one forth of multipliers when using two's complement. Regardless of the filter performance, using SMPT_2 representation cost the least multiplication size. Implementing a 12 bits FIR filter using SPT_4 and CSD representation all can meet the desired specification, but SPT_4 cost less multiplication size. Hence, in order to measure the filter performance and computation speed together, we construct a cost function to estimate the cost in Chapter 5.

Chapter 5

COST FUNCTION

In Chapter 3, we gave the quantization error distribution of each number system and also introduced implementing the FIR LPF using different number system. We note that the round-off error is caused by the quantizing the coefficients using different number system, which will also influence the filter performance. Furthermore, the multiplication size of number of multipliers are related to the number representation as well.

In this chapter, we will proposal a cost function which can combine the filtering performance and computation cost together for estimating the total cost. The cost function will be detailed in Section 5.2. This cost function is used specially for the representation of CSD and SPT$_K$. We give the analysis and comparison of the cost for these two number representation systems.

5.1 ERROR COST AND COMPUTATION COST

Figure 5.1 shows the parameters of a FIR LPF. There are passband ripple, worst attenuation, cut-off frequency and stopband edge frequency. These specifications should be considered when we are implementing designed filter with different number system. In the actual implementation, the error will occur on F_c, F_{st} as well as worst attenuation W. The E_c, E_{st} and E_w are the error

Table 5.1: Specification and error of implementing 8 bits FIR low-passfilter using CSD representation

Parameters	Desired Specification	Actual Specification	Error
F_c	0.4167	0.3948	0.0219(E_c)
F_{st}	0.5	0.4980	0.0020(E_{st})
W	-40	-37.71	2.29(E_w)

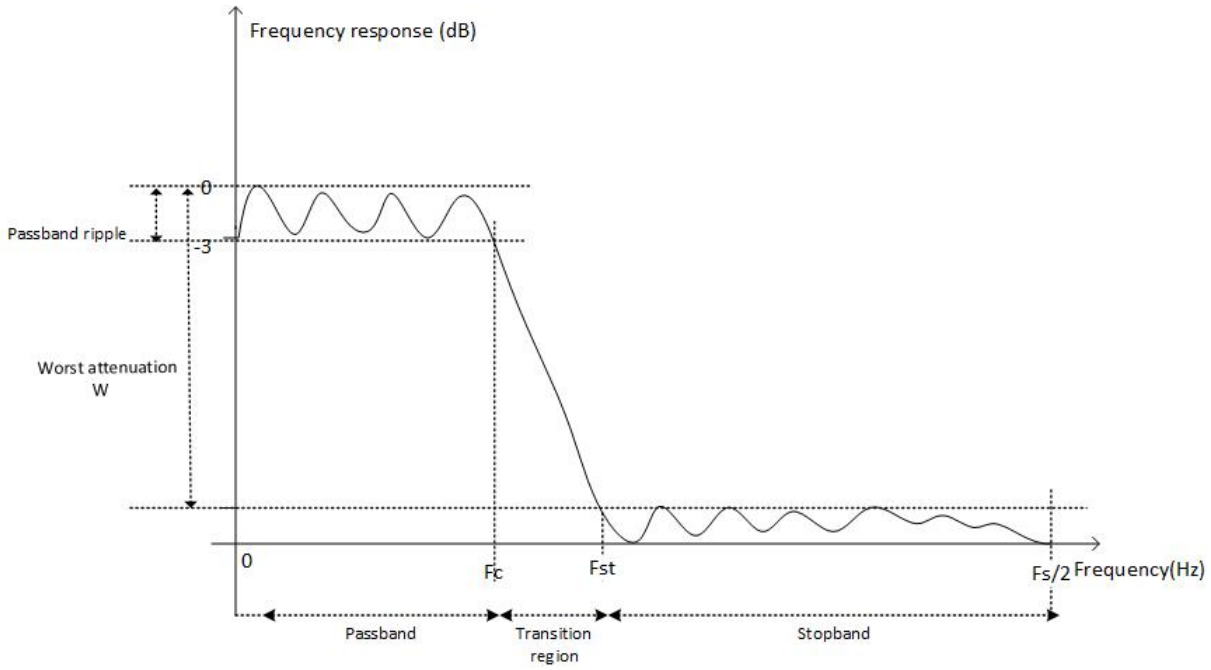


Figure 5.1: Frequency response of FIR LPF

on the cut-off frequency, stop-band frequency and worst attenuation respectively, it is shown in the Table 5.1. The error is the absolute value between the actual specification and designed specification.

The filter performance is presented by the error and the filter computation is measured by the multiplication size. The error can be reduced by increasing the word-length and filter length but the computation will increase. Therefore, it is important to balance the filter performance and computation to choose a relatively higher performance and lower computed digital filter. In this way a proper cost function should contain the error cost and computation cost together. One the one hand, the error cost can clearly reflect the filter performance of the implemented filter, lower cost indicates the better filter performance. On the other hand, the smaller the computation cost, the faster the computing speed.

The error cost is related to F_c, F_{st} and W. Meanwhile, the computation cost is related to multiplication size M. The error cost is as below,

$$E = \frac{E_c}{\tau_c} + \frac{E_{st}}{\tau_{st}} + 10^{(E_w/20)}, \tag{5.1}$$

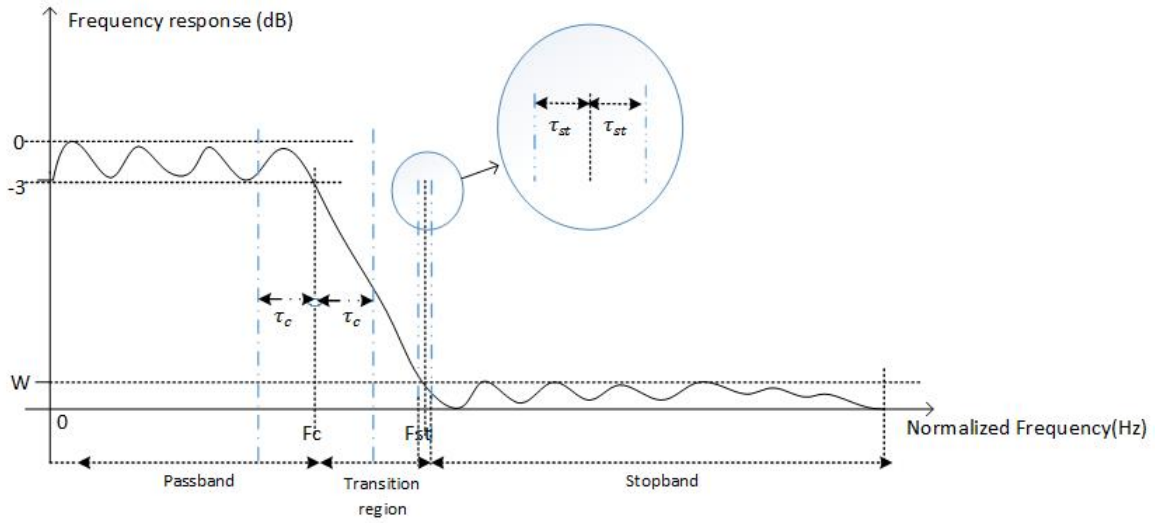


Figure 5.2: The parameters used cost function

where τ_c denotes the tolerated error on the cut-off frequency, while τ_{st} denote the tolerated frequency error on the stopband edge frequency, which are shown in Figure 5.2. Normalized frequency and magnitude values are employed to help eliminating the unit. This error cost equation depicts that the E_c and E_{st} are compared with the tolerated error value. The minimum E is 1, when E_c, E_{st} and E_w all equal to 0.

The cost of multipliers is the only component that contributes to the computation cost. N_m denotes the multiplication size used in implementation. The computation cost is,

$$C = \frac{N_m}{M}, \tag{5.2}$$

where M is the average multiplication size which can keep the worst attenuation at W. The ratio of actual multiplication size and the average multiplication size is used to measure the computation cost. It is clear that the smaller the value C is the faster overall computation. The best computation is when C equals to 0, which means there are 0 multipliers is used in implementing filter, which is impossible in a practical way. It is necessary to limit the acceptable maximum multiplication size, which is $2M$. If the C is larger than the 2, the value of C will exceed our acceptable number of multipliers. As a result, keep the value of C between 0 to 2 is the acceptable option in practice.

5.2 COST FUNCTION

The cost function combines the error cost and computation cost together. The cost function is,

$$C_f = E + \delta C = \frac{E_c}{\tau_c} + \frac{E_{st}}{\tau_{st}} + 10^{(E_w/20)} + \delta \frac{N_m}{M}, \tag{5.3}$$

where δ is a differential weighting function. However, no matter how the value of δ is to be chosen, the best situation for cost is always C_f equal to 1, when the E_c, E_{st}, E_w and N_m are all 0. Applying this cost function in implementing the FIR LPF, the cost value of filter with the best performance and quickest computation should be as close as possible to 1. A preliminary sensitivity analysis, $\delta = 1$ and $\delta = 3$, indicates the selection of the test value for δ is not crucial to the total cost further. One example is shown in Appendix A. For our examples, we choose $\delta = 1$.

5.3 DESIRED FIR LPF

The specification of desired FIR LPF is as below:

1. Sampling frequency : $F_s = 2$ (Normalized Hz)
2. Cut-off frequency : $F_c = 0.4167$ (Normalized Hz)
3. Tolerated error in cut-off frequency: $\tau_c = 0.0417$ (Normalized Hz)
4. Stopband frequency edge: $F_{st} = 0.5$ (Normalized Hz)
5. Tolerated error in stopband freqency edge: $\tau_{st} = 0.0042$ (Normalized Hz)
6. Worst attenuation : $W = -40(dB)$
7. Average multiplication size: $M = 40$

Figure 5.3 gives all the specifications of the desired filter. These specifications is used as parameters in the cost functions. We applied the above cost function to 247 FIR LPF for each representations to find the most approximate FIR LPF for desired filter. Various word-length is

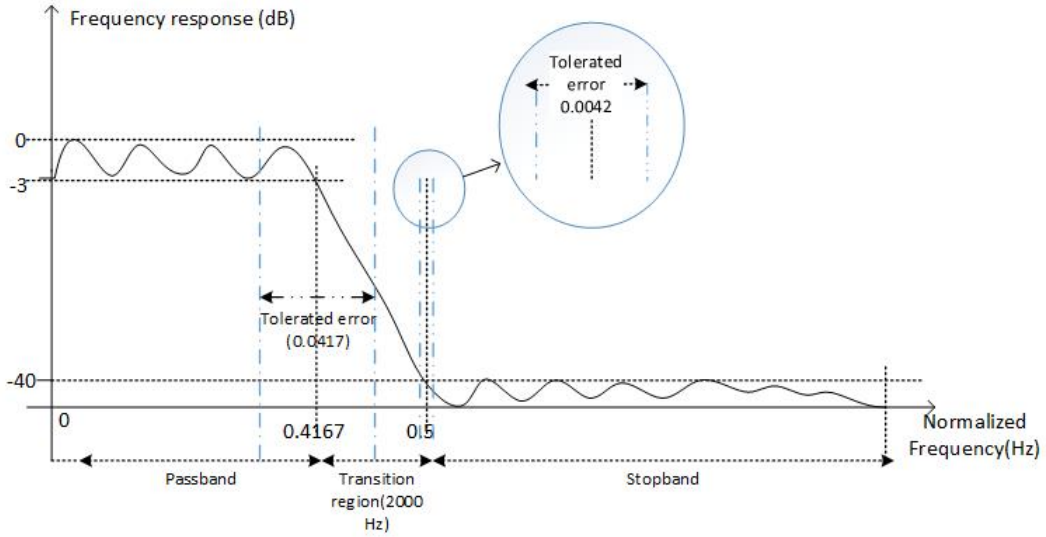


Figure 5.3: The frequency response of desired FIR LPF

chosen from 8 bits to 18 bits, in addition 6 bits and 32 bits are considered as special situation. The filter length is chosen from all even numbers between 16 to 52.

5.4 ANALYSIS

This section will show the analysis of cost using CSD number representation, SPT_K number representation and SPT_K with CSD constraint representation. The error cost and computation cost is also analyzed for all representations. For SPT_K number representation and SPT_K with CSD constraint(SPTCSD_K) representation, the analyses are given when the K equals to 2, 3, 4 respectively. Furthermore, some comparison are made as well. The data used for analysis and comparisons are come from the 1159 approximate filter used above number representations at various word-length and filter length. As the data used to calculate the cost is numerous for 1159 approximate filters, so we just give one set of data as examples in the Appendix B to shown the parameters and calculations got from trials. The cost of the rest of trials we directly give in the Appendix C.

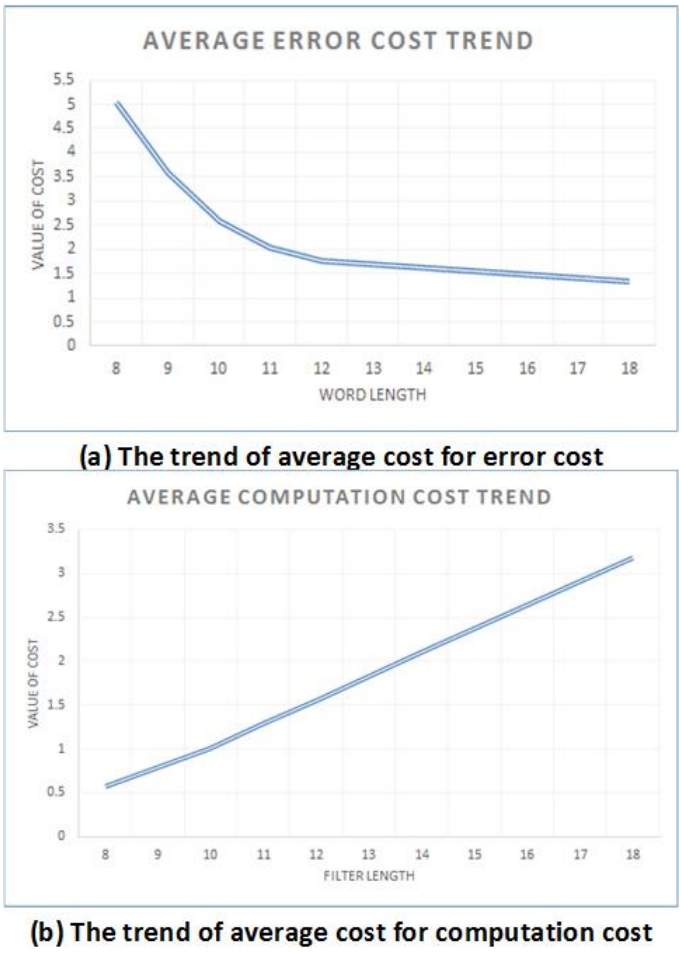
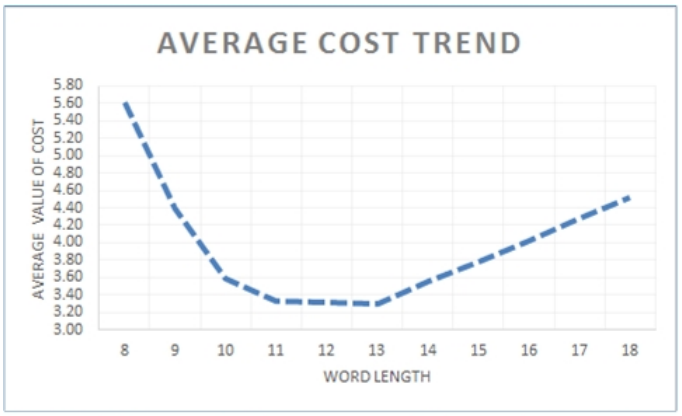


Figure 5.4: The averagec error cost and computation cost for CSD representation

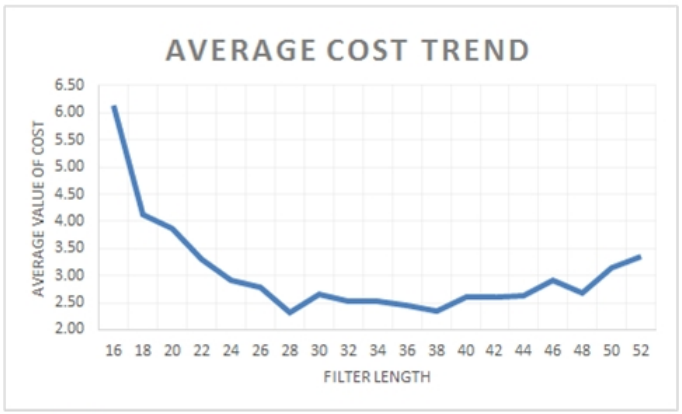
5.4.1 Analysis of the Cost of Using CSD Number Representation

Figure 5.4 gives the error cost and computation cost using CSD representation. Figure 5.4(a) illustrates the error cost constantly decreases with increasing of word-length. In contrast, the computation cost has a steady increasing with growing of the word-length in Figure 5.4(b). It indicates the computation cost will influence the cost dramatically with the increasing of the word-length. Conversely, the error cost gives more influences on the cost when the word-length is small.

Figure 5.5 shows the average cost for word-length and filter length using CSD representation. From Figure 5.5(a), it is obviously that the cost is a U curve, the optimal word-length is 11 bits. For the filter length, the lowest cost is when the filter length is 28 in Figure 5.5(b).



(a) The trend of average cost for word-length



(b) The trend of average cost for filter length

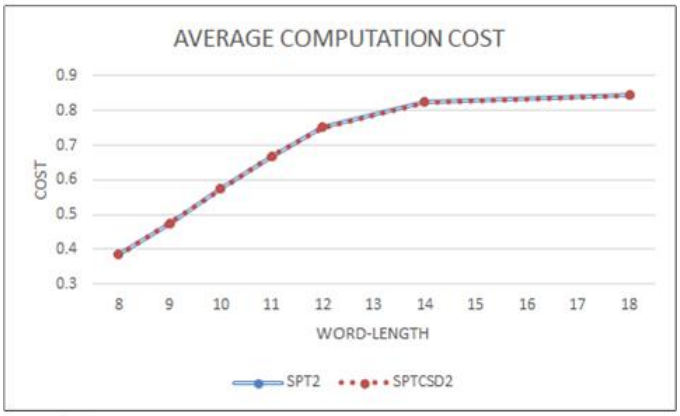
Figure 5.5: The average cost using CSD representation

Above all, in order implement the desired FIR LPF using CSD representation, the best choice is when the word-length is 11 bits and the filter length is 28.

5.4.2 Analysis of the Cost Using SPT_2 Number Representation and $SPTCSD_2$ Representation

Figure 5.6 gives average error cost and computation cost using SPT_2 representation and $SPTCSD_2$ representation. The error cost and computation cost of both of the representations are all the same. In Figure 5.6(a), the error cost has a significantly drop before 10 bits, but keep steady after that. The computation cost in Figure 5.6(b) increases quickly before 14 bits. However, after the word-length is greater than 14 bits, the computation cost remains at the same value. From values of error cost and computation cost, it is worth noting that the error cost dominate the cost.

(a) Average error cost for word-length

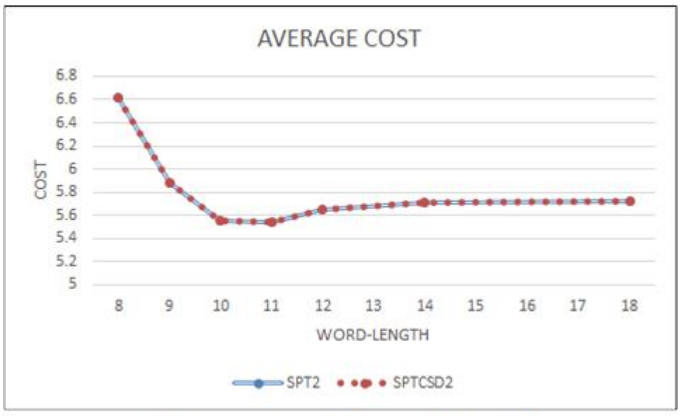


(b) Average computation cost for word-length

Figure 5.6: The average error cost and computation cost for word-length SPT_2 representation and $SPTCSD_2$ representation

The Figure 5.7 illustrates the average cost trend both for word-length and filter length using SPT_2 representation and $SPTCSD_2$ representation. It is clearly that the when the K = 2, the cost of SPT_2 representation and $SPTCSD_2$ representation are the same. Figure5.7(a) shown that 10 bits is the optimal word-length to implement a FIR LPF filter and the cost keep steady after 12 bits for both of representations. Figure 5.7(b) gives the average cost for filter length, the trend is a rough U curve and it indicates that when the filter length is 28, the cost is minimum for both representations.

When the $K = 2$, it is obviously that the either the cost, error cost or computation cost are all the same for using SPT_2 representation and $SPTCSD_2$ representation. The approximate FIR LPF using SPT_2 representation and $SPTCSD_2$ representation is when the word-length is 10 bits and the filter length is 28.



(a) Average cost for word-length

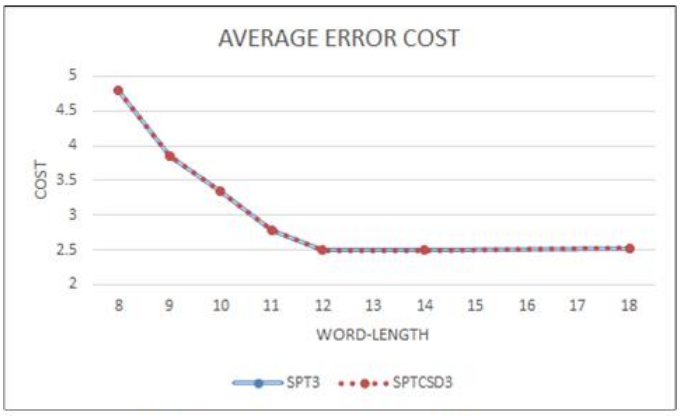
(b) Average cost for filter length

Figure 5.7: The average cost for word-length and filter length using SPT_2 representation and $SPTCSD_2$ representation

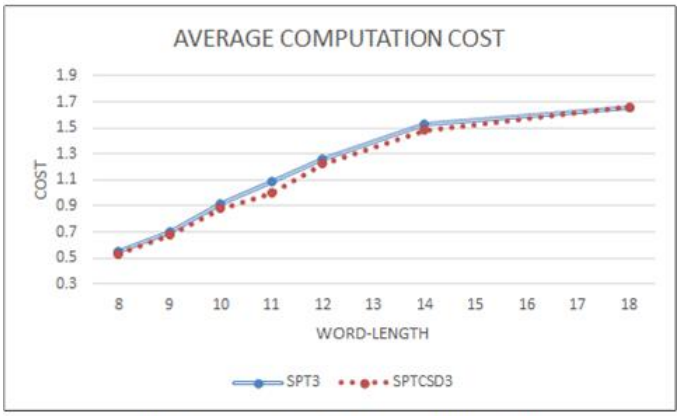
5.4.3 Analysis of the Cost Using SPT_3 Number Representation and $SPTCSD_3$ Representation

Figure 5.8(a) shows that the error cost for SPT_3 representation and $SPTCSD_3$ representations are superposition. The error cost can be decreased by increase the word-length. Although increasing of word-length after 12 bits, the error cost cannot be decreased anymore. On the contrary, the computation costs for SPT_3 representation and $SPTCSD_3$ representation are both increasing with the increasing of word-length in Figure 5.8(b). The cost for SPT_3 representation is a little larger than using $SPTCSD_3$ representation.

In Figure 5.9(a) shows the cost for word-length is a U curve and the optimal point is when the word-length is 12 bits for SPT_3 representation and $SPTCSD_3$ representation. The cost for SPT_3 representation is a slight higher than the cost for using $SPTCSD_3$ representation. In Figure



(a)Average cost for word-length

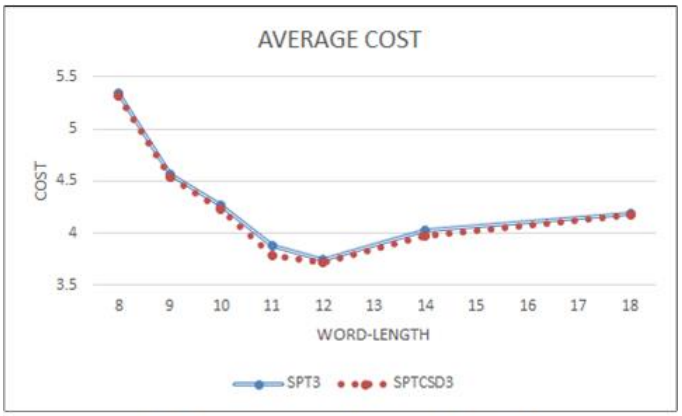


(b) Average cost for word-length

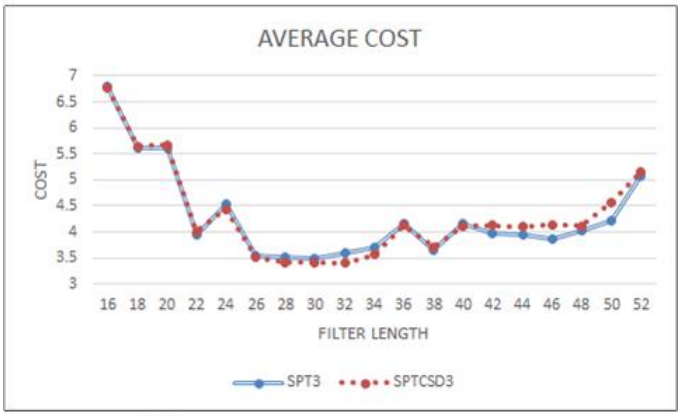
Figure 5.8: The average error cost and computation cost for word-length SPT_3 representation and $SPTCSD_3$ representation

5.9(b), the cost for SPT_3 representation and $SPTCSD_3$ representation has a minor different but their trend is still the same. In general, the cost decreases when the filter length smaller than 26, then has a slowly increasing.

When K =3, the results show that the error cost for SPT_3 representation and $SPTCSD_3$ representation are still the same. The cost for word-length using $SPTCSD_3$ representation is slight lower than using SPT_3 representation due to the lower computation cost of using $SPTCSD_3$ representation. Over all, the best choice for implement the FIR LPF when the word-length is 12 bits while the filter length is 32 coefficients.



(a) Average cost for word-length



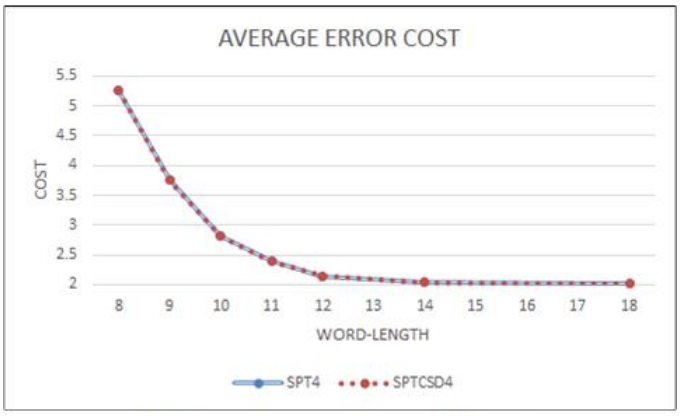
(b) Average cost for filter length

Figure 5.9: The average cost for word-length and filter length using SPT_3 representation and $SPTCSD_3$ representation

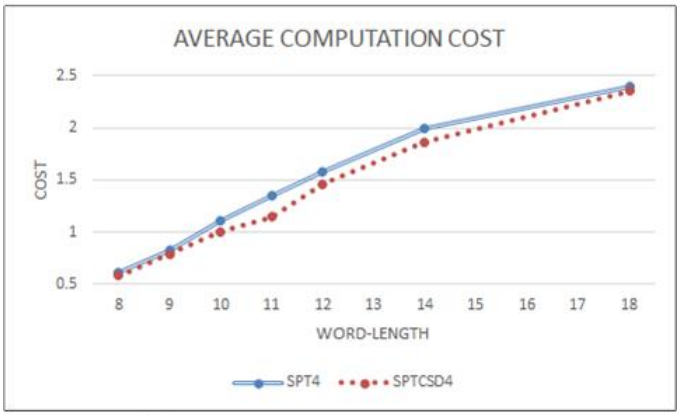
5.4.4 Analysis of the Cost Using SPT_4 Number Representation and $SPTCSD_4$ Representation

Figure 5.10(a) illustrates the error cost for word-length. The error cost is same for these two presentations. Their error costs have a significant drop before 12 bits, after that the cost just slight decreases. In contrast, computation cost for SPT_4 representation and $SPTCSD_4$ representation are constantly growing with increasing of word-length in Figure 5.10(b). However, the computation cost for SPT_4 representation is greater than that for $SPTCSD_4$ representation.

It is clearly that the cost for using SPT_4 representation is higher than it is using $SPTCSD_4$ representation in Figure 5.11(a). The cost is a U shape curve and its minimum cost is when the word-length is 11 bits for both of representations. In general, the cost in Figure 5.11(b) is also a rough U curve for both of the representations. Their cost decreases when the filter length



(a)Average cost for word-length

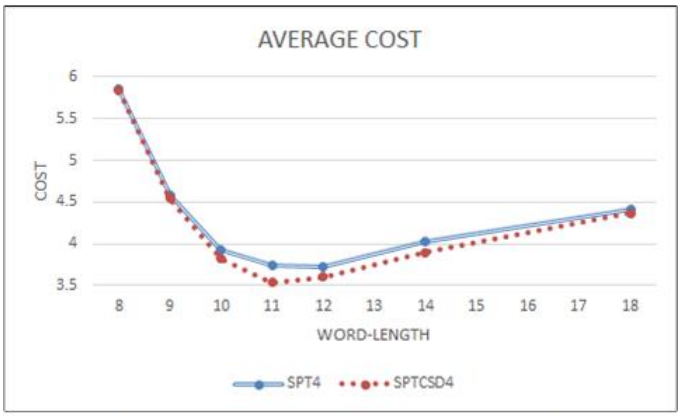


(b) Average cost for word-length

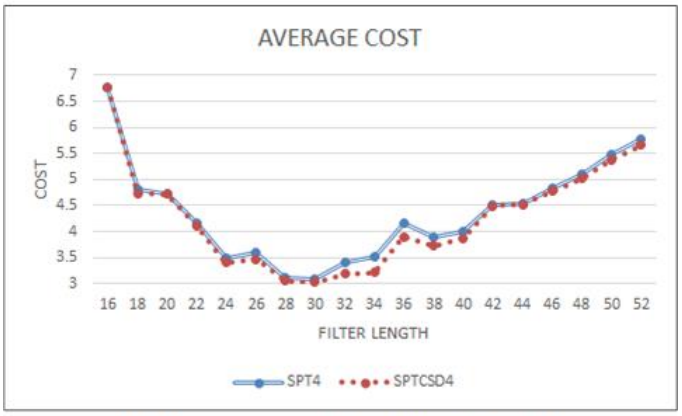
Figure 5.10: The average error cost and computation cost for word-length SPT_4 representation and $SPTCSD_4$ representation

increase from 16 to 30 but after that it tend to increase again. The optimal filter length is 28. The cost for SPT_4 representation and $SPTCSD_4$ representation do not have too much different for filter length.

Above all, when the K=4, the cost for $SPTCSD_4$ representation is lower than using SPT_4 representation because as their error cost are all the same but the computation $SPTCSD_4$ representation is lower. The computation cost gives more weight to the cost with increase the value of K. When the word-length is 11 bits and filter length is 28, the approximate FIR LPF will have smallest cost.



(a) Average cost for word-length



(b)Average cost for filter length

Figure 5.11: The average cost for word-length and filter length using SPT_4 representation and SPTCSD_4 representation

5.5 SIMULATION RESULT

According to the analysis of Section 5.4, we got the simulation results for the approximate FIR LPF using each representations.

For CSD representation, Figure 5.12 gives the frequency response of the implemented FIR LPF. It is can be seen that the cut-off frequency, stopband edge frequency and the stopband peak ripple are all very close to the desired filter. The multiplication size is 46.

As the error cost for SPT_K representation or SPTCSD_K representation are all the same, so their frequency response for the same filter are the same as well.Figure 5.13 illustrates the frequency responses of FIR LPF using SPT_2 representation or SPTCSD_2 representation. The

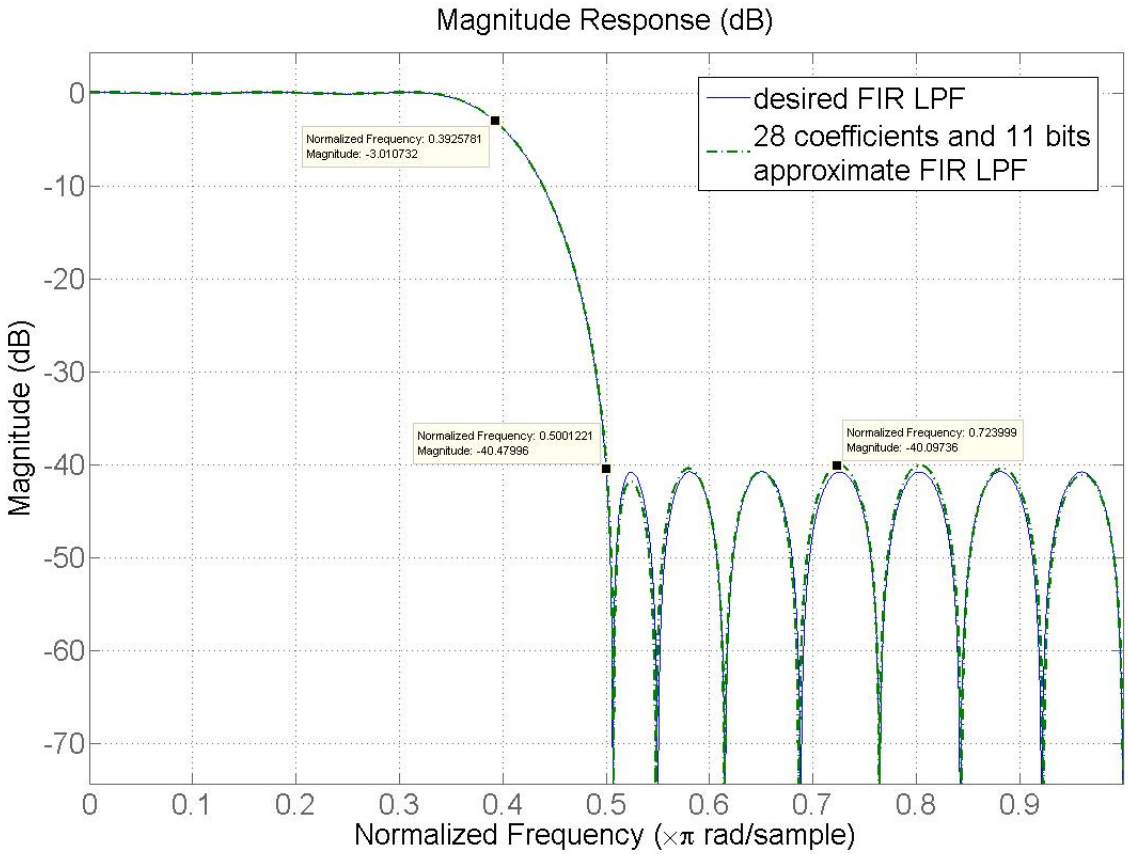


Figure 5.12: The frequency response of FIR LPF using CSD number representation (word-length is 11 bits and filter length is 28)

cut-off frequency, stopband edge frequency and the worst attenuation of approximate filter meet the requirements of the specification. The multiplication size for using SPT_2 representation or $SPTCSD_2$ representation are all 22.

Figure 5.14 shows the frequency response of implemented approximate FIR LPF when the K = 3. The specifications for approximate filter are all meet the desired requirements. The multiplication size of approximate FIR LPF for using either SPT_3 representation or $SPTCSD_3$ representation is 54.

Figure 5.15 shows the frequency response of approximate FIR LPF is very similar to the desired FIR LPF. No matter of cut-off frequency, stopband edge frequency or worst attenuation are all very close to the desired specifications. The multiplication size for SPT_4 representation is 54 but for $SPTCSD_4$ representation is just 50.

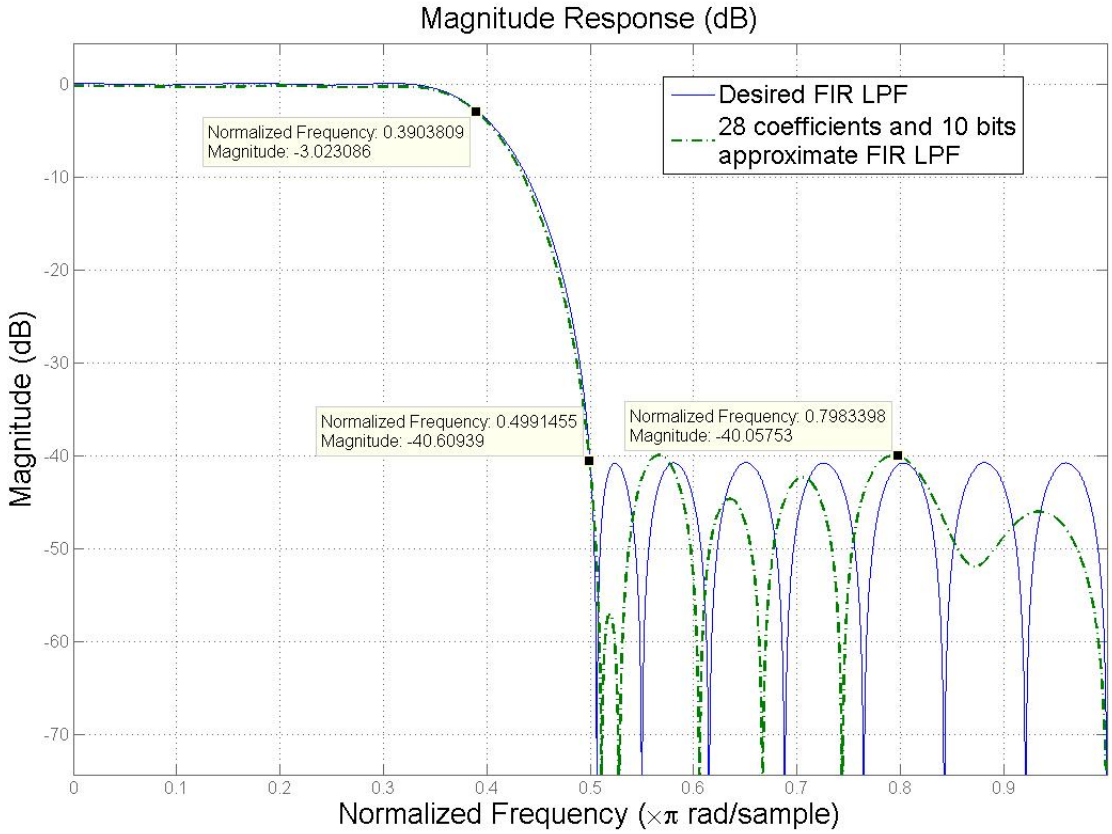


Figure 5.13: The frequency response of FIR LPF using SPT_2 representation or SPTCSD_2 representation(word-length is 10 bits and filter length is 28)

5.6 COMPARISON

In this section, a brief of comparison of the cost of for using CSD number representation , SPT_K and SPTCSD_K representation will be given. Comparison data are still using the cost for all approximate implemented FIR LPF.

5.6.1 Comparison of SPT_K and SPTCSD_K Representation

SPT_K and SPTCSD_K representation has different computation cost effects but for the same K, their error cost are the same.

Figure 5.16(a) gives the cost of SPTCSD_K and SPTCSD_K representation when the K is 2, 3 and 4. We already know that when the $K = 2$, the cost of SPT_2 and SPTCSD_2 representation

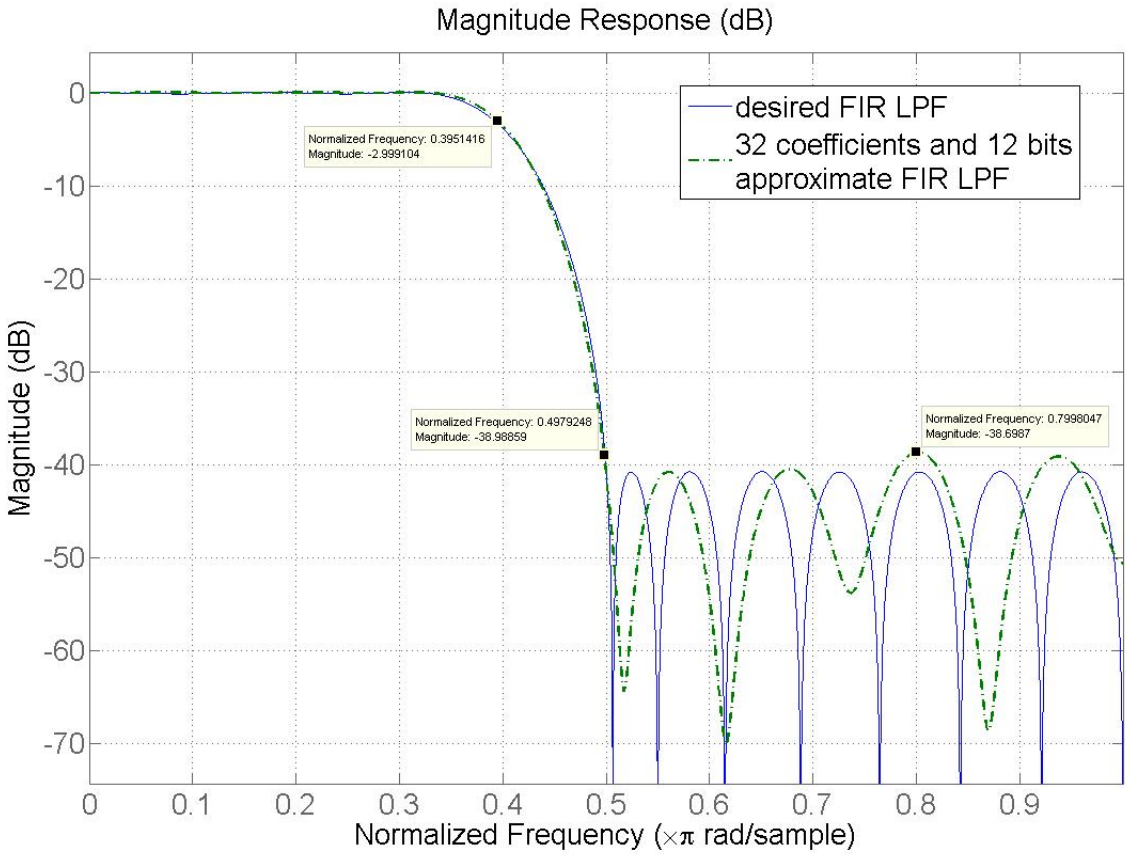


Figure 5.14: The frequency response of FIR LPF using SPT_3 representation or $SPTCSD_3$ representation(word-length is 12 bits and filter length is 32)

are the same. However, in Figure 5.16, it is clearly that the cost for $K = 2$ is much higher than others. When the K is 3, the cost for SPT_3 representation and $SPTCSD_3$ representation do not have much difference but their cost is slight higher than when the $K = 4$. The lowest cost is using $SPTCSD_4$ representation.

In Figure 5.16(b), the highest cost for filter length is when the $K = 2$, but some unique word-length has very low cost due to the quantization effects for the coefficients, like when the word-length is 28 using SPT_2 still can get very small cost. In general, the cost decreases with increasing the value of K. The cost for $SPTCSD_4$ is lower than others before the word-length is smaller than 40 and it starts increasing after that. The reason is the constantly growing computation cost with increasing of word-length.

Figure 5.17(a) is the error cost using SPT_K and $SPTCSD_K$ representations when the K is

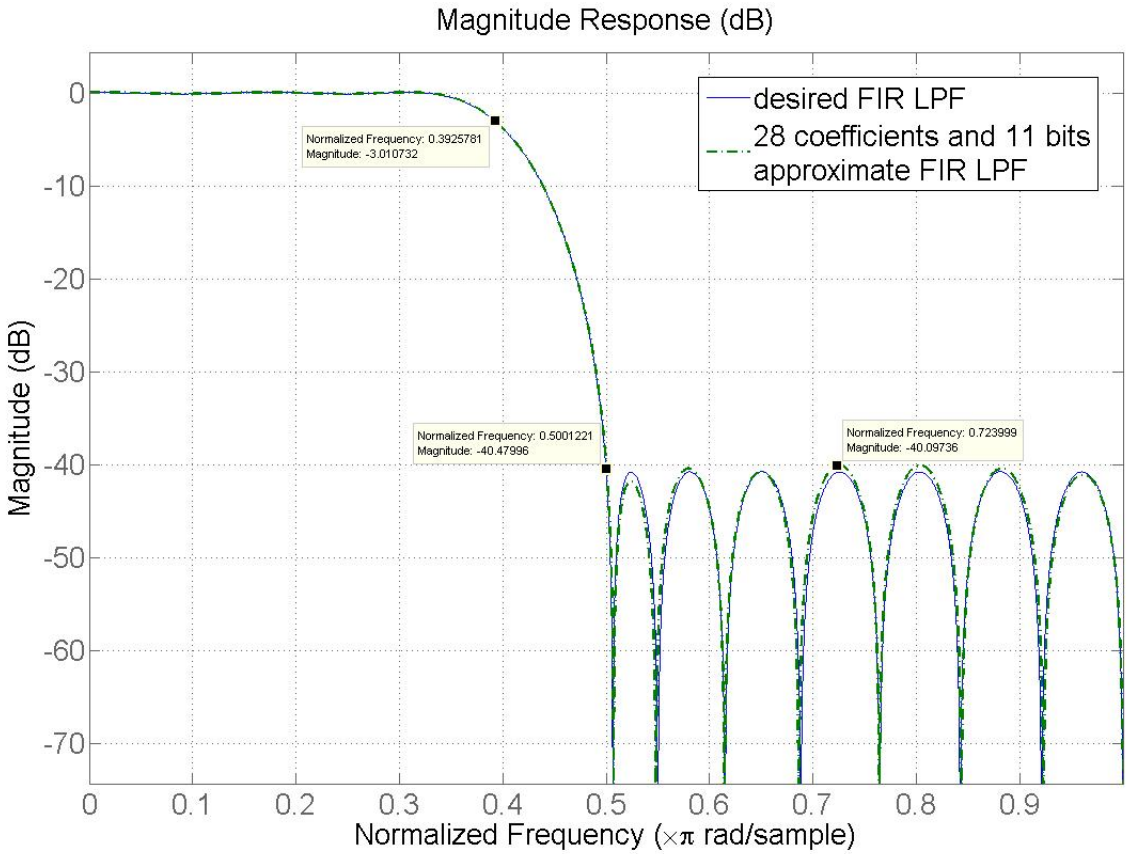


Figure 5.15: The frequency response of FIR LPF using SPT_4 representation or SPTCSD_4 representation(word-length is 11 bits and filter length is 28)

different value. It is obviously, with increase of the K, the error cost drops for the same word-length. Figure 5.17(b) illustrates that the computation cost are increased with increase of K for the same word-length and the cost for using SPT_4 representation is the highest.

5.6.2 Comparison of CSD Representation and SPTCSD_K Representation

The error cost for CSD representation drops all the time with the growing word-length. This is much different with the SPTCSD_K representation.

In Figure 5.18, it is obviously that the cost of using representation of SPTCSD_3 and SPTCSD_4 are quite close to the CSD representation, whereas the cost for SPTCSD_2 is much higher than the rest of representations. It is worth noting that the cost for SPTCSD_K representation is

(a) Average error for word-length

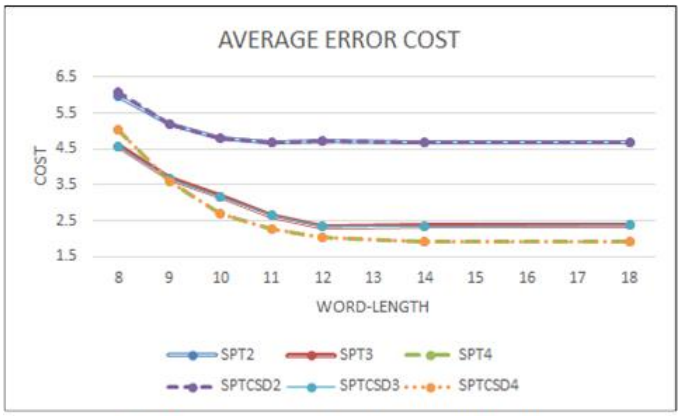
(b) Average error for filter length

Figure 5.16: The average cost for word-length using SPT_K representation and SPTCSD_K representation

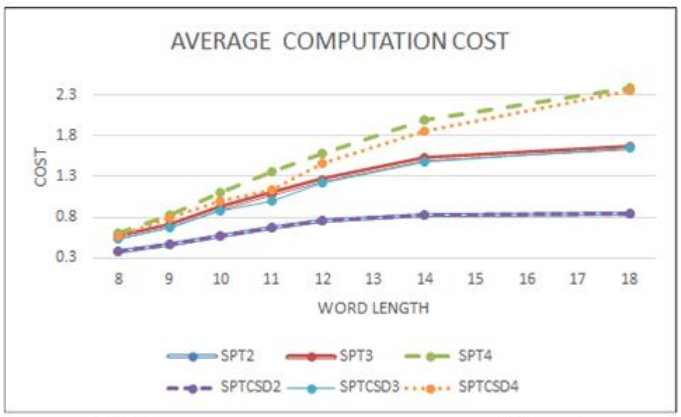
smaller than CSD representation after the 16 bits.

With increasing of the value of K, the error cost of SPTCSD_K representation tends to closer to the cost of CSD number representation system, which are shown in the Figure 5.19(a). It is obviously that when the word-length is 8 bits, the error cost of SPTCSD_4 are the same with the CSD representation. This results is agree with the results in Chapter 3, Section 4.7 that when the K equal and greater than the half of given word-length, the SPT_K becomes CSD representation. However, the error cost cannot be decreased after 9 bits for SPTCSD_4 representation due to the limited K value.

The Figure 5.19(b) shows the computation cost for the word-length, the CSD computation cost is a straight constantly growing line. The computation cost of SPTCSD_4 is smaller than CSD



(a) Average error cost for word-length



(b) Average computation cost for word-length

Figure 5.17: The average error cost and computation cost for word-length using SPT_K representation and SPTCSD_K representation

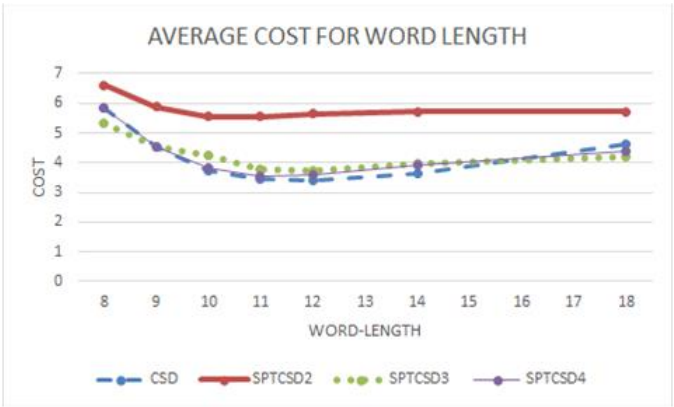
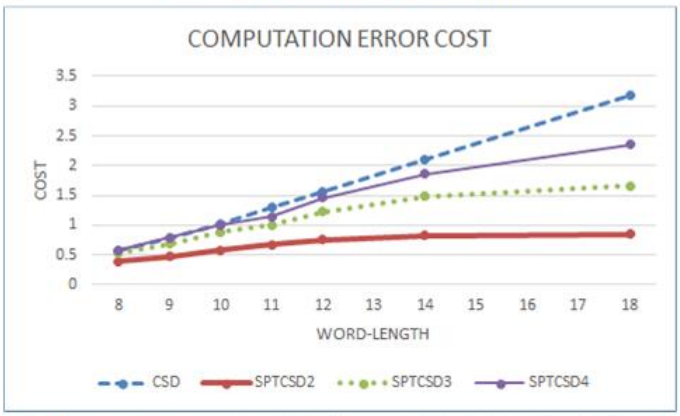


Figure 5.18: The average cost for word-length using CSD number representation and SPTCSD_K representation

(a) Average error cost for word-length using CSD and SPTCSDK number representation



a) Average computation cost for word-length using CSD and SPTCSDK number representation

Figure 5.19: The comparison of CSD representation and SPTCSD$_K$ representation

representation when the word-length is larger than 10 bits. When the K=2 and K =3, the computation cost for SPTK with CSD constraint is half and one-third of the computation cost using CSD representation, respectively.

Overall, with increasing of value of K, the average error cost for SPTCSD$_K$ representation is decreased and the computation cost increases but the increased speed is not as quickly as CSD representation. The totally cost for SPTCSD$_4$ is even smaller than CSD representation after the word-length is greater than 16 bits. It indicates that using SPTCSD$_K$ implementing N bits FIR LPF filter is a more efficient way when $N/3 \leq K < N/2$.

5.7 SUMMARY

In this chapter, the cost function is derived from experiments to suggest optimal word-length and filter length for different number representations. This cost function straight forward used absolute error to measure the error cost and computation cost and then made them up together.

The analysis illustrates that the when the K is equal to 2, the error cost of SPTCSD_2 representation is keeping constant line after 10 bits which is agree with the results in Chapter 4 that the error cost cannot be decreased after 10 bits using SPTCSD_2 representation. With the value of K is increasing, the cost of SPTCSD_K representation is more and more close to the cost using the CSD representation. When the K is up to 4, cost of SPTCSD_K is the better than CSD representation after 16 bits, which indicates that for implementing a N bits FIR LPF filter using SPTCSD_K when $N/3 \leq K < N/2$ to represent the coefficients is better than using CSD representation.

In this chapter, we already showed the ratio of the error cost and computation cost but the error cost still contain three components and it very difficulty to test all these three components sensitivities. Therefore, this is a topic for the future work.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSION

In this thesis, we studied two pieces of the work. The first part is implementing the FIR LPF with number representation of two's complement, CSD, SMPT_K, SPT_K and SPTCSD_K and looked into the quantization effects of using all these four number representations, we concluded as follows:

- The round-off error quantization distribution for two's complement number representation and CSD number representation are all uniform distribution [1].

- In SPT field, we looked at the statistical measure of the round-off error when converting each number into SPT term. The SMPT_K number representation is introduced and we derived the formula of probability density distribution of SMPT_2 number representation which demonstrate as a staircase profile due to its nonuniform quantized steps after 4 bits. The formula is given by Equation 4.4 and Equation 4.5.

- We simulated the trend of SPT_K number representation when the K is equal to 2, 3 and 4. The formula of the probability density distribution of round-off error for SPT_2 is also given in Equation 4.7 and Equation 4.8, which is also shown as the staircase profile when the word-length greater than 5 bits.

- The quantized steps of SPTCSD_K number representation is the same as the SPT_K number representation when the representable value range is [-0.5 0.5]. Within this range, the

SPTCSD$_K$ number representation share the same probability density distribution formula and trend with SPT$_K$ number representation.

- The K is crucial value in SPT field. The SPTCSD$_K$ representation becomes the CSD representation when the K increases to the half of the given word-length. In this way, the quantization step tends to uniform step with growing of K. However, when the K is much smaller than given word-length, the round-off error cannot decrease by increasing the word-length after certain bits. For example, when the $K = 3$, the round-off error cannot be decreased when the word-length greater than 12 bits.

In the second part of work the cost function is constructed to indicate the proper word-length and filter length in order to implement desirable FIR LPF using different number representation system. We have:

- Successfully found the proper word-length and filter length for approximate FIR LPF using CSD representation, SPT $_K$ representation and SPTCSD$_K$ representation to achieve the desired specifications when the K equals to 2, 3, 4. The simulation results is shown in Figure 5.12, Figure 5.13, Figure 5.14 and Figure 5.15.
- Analyzed the cost from 1159 approximate FIR LPFs which use our cost function shown in Equation 5.3. The comparison of cost for CSD representation and SPTCSD$_K$ representation turned out the conclusion which we proposed in first part that when the K is much smaller than the given word-length, the round-off error become stable after the certain bits.
- Proved that the quantization error for SPT $_K$ representation and SPTCSD$_K$ representation are the same due to all the coefficients of the FIR LPF are all in the range [-0.5 0.5].
- Indicated that for implementing a N bits FIR LPF, the SPTCSD$_K$ representation is better choice than CSD representation when the $N/3 \leq K < N/2$.

6.2 FUTURE WORK

There are some aspects that we want to do more in the future:

- It is necessary to prove error distribution formula for number representation of SMPT_2 and SPT_2 by the mathematically.
- It is worth to derive a general distribution formula for SPT_K number representations.
- The error cost(E) in cost function (Equation 5.3) contains three components, the sensibility of each component on the cost function has yet to be investigated.

Appendix A

AN EXAMPLE OF CHOOSING δ =1 AND δ =3

For the cost function, the δ is a differential weighting function. Here is the example that when the δ =1 and δ=3 used in the cost function and then apply this cost function to the approximate FIR LPF used CSD number representation when the word-length is 8 bits. The cost of C_f is in the Table A.1.

Table A.1: The cost for filter length when the δ=1 and δ=3

Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
$C_f(\delta$=1)	6.967	4.862	4.796	4.258	3.630	3.581	3.152	3.295	3.372	3.447	3.597	3.750	3.733	4.012	4.163	4.380	4.473	4.577	5.051
$C_f(\delta$=3)	9.092	7.200	6.996	7.145	6.255	6.469	6.189	6.583	6.910	6.884	7.009	7.337	7.246	7.487	7.825	8.405	8.435	8.689	9.551

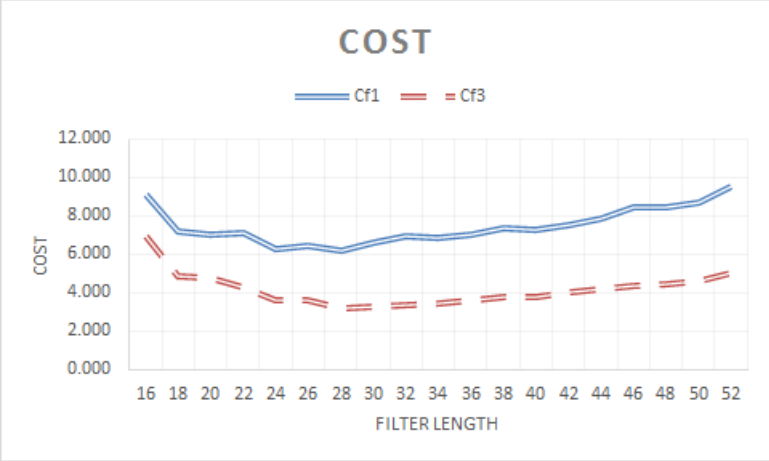


Figure A.1: The comparison of cost when the δ =1 and δ=3

It is obviously that the ratio δ changes from 1 to 3 results the total cost increases but the trend of cost for filter length still keeps the same. It indicates that this cost function is not sensitively with changing of δ.

Appendix B

THE EXAMPLES OF COLLECTED PARAMETERS FOR COST FUNCTION

Here is a set of collected parameters are used for calculating the cost. Table B.1 and Table B.2 shows the cut-off frequency, stopband edge frequency, worst attenuation and the multiplications size of the 8 bits approximate FIR LPF used CSD representation with different filter lengths. The roll-off and overshoot is used to double check the filter performance. The error cost (E) is got by applying the absolute error into the Equation 5.1. Computation cost(C) are calculated by feeding multiplication size into Equation 5.2. The error cost and computation cost are calculated separately are used for the later analysis and comparison. At last, we get the cost(C_f) for each filter by Equation 5.3.

It is obviously for implementing an 8 bits FIR LPF using CSD representation, the lowest cost is when the filter length is 28. This is just one piece of data we got for analysis the for CSD representation. For whole work, it is necessary to obtain all the error cost and computation cost to analysis and comparison.

Table B.1: The parameters of 8 bits FIR LPF with different filter length (16-32)

Parameters \ Filter Length	16	18	20	22	24	26	28	30	32
cutoff	9240.23	9339.84	9310.55	9392.58	9363.28	9427.73	9439.45	9427.73	9474.61
stopband	11976.56	11906.25	11929.69	11876.95	12017.58	11882.81	12000.00	11947.27	11953.13
worst attenuation	-26.40	-29.66	-29.01	-32.41	-36.40	-36.40	-39.08	-36.79	-37.71
roll off	207.70	252.82	241.61	288.55	308.12	332.30	346.11	328.49	343.90
overshoot	0.40	0.23	0.22	0.14	0.18	0.14	0.14	0.04	0.08
multiplication	20	22	20	26	24	26	26	26	26
cutoff error (E_c)	759.77	660.16	689.45	607.42	636.72	572.27	560.55	572.27	525.39
stopband error(E_{st})	23.44	93.75	70.31	123.05	17.58	117.19	0.00	52.73	46.88
attenuation cost (E_w)	0.05	0.03	0.04	0.02	0.02	0.02	0.01	0.01	0.01
computation(N_m)	20	22	20	26	24	26	26	26	26
error cost (E)	6.31	5.25	5.33	4.50	2.49	3.43	1.80	2.71	2.44
computation cost (C)	0.5	0.55	0.5	0.65	0.6	0.65	0.65	0.65	0.65
cost(C_f)	6.814	5.803	5.830	5.150	3.095	4.076	2.446	3.358	3.091

Table B.2: The parameters of 8 bits FIR LPF with different filter length (34-52)

Parameters \ Filter Length	34	36	38	40	42	44	46	48	50	52
cutoff	9468.75	9480.47	9492.19	9521.48	9539.06	9539.06	9544.92	9556.64	9568.36	9568.36
stopband	11853.52	11783.2	11648.44	11507.81	11501.95	11501.95	11431.64	11496.09	11490.23	11507.81
worst attenuation	-38.32	-34.1	-35.73	-35.03	-36.19	-36.19	-36.19	-36.02	-36.93	-33.77
roll off	362.06	329.37	368.18	389.26	408.4	408.4	423.73	411.45	426.83	383.88
overshoot	0.13	0.1	0.13	0.1	0.12	0.12	0.13	0.18	0.12	0.13
multiplication	24	24	22	22	22	22	22	22	22	22
cutoff error (E_c)	531.25	519.53	507.81	478.52	460.94	460.94	455.08	443.36	431.64	431.64
stopband error(E_{st})	146.48	216.8	351.56	492.19	498.05	498.05	568.36	503.91	509.77	492.19
attenuation cost (E_w)	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02
computation(N_m)	24	24	22	22	22	22	22	22	22	22
error cost (E)	3.34	4.88	5.84	7.37	7.16	7.16	7.86	7.24	7.11	7.63
computation cost	0.6	0.6	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
cost(C_f)	3.944	5.478	6.39	7.919	7.714	7.714	8.411	7.79	7.662	8.18

Appendix C

THE DATA USED FOR ANALYSIS

This appendix gives the costs of 1159 approximate FIR LPFs used different number representations at various word-lengths and filter lengths.

Table C.1: The cost of FIR LPF using CSD number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	average
6 bits	7.48	5.21	5.21	9.87	9.87	9.87	12.07	9.87	12.07	12.07	12.07	17.62	17.62	11.97	11.97	17.49	17.49	17.49	17.49	12.36
8 bits	6.28	5.44	5.44	4.88	2.93	3.91	2.32	3.20	2.95	3.81	5.26	6.21	7.72	7.54	7.54	8.24	7.61	7.50	7.95	5.62
9 bits	6.28	4.13	3.87	4.33	3.83	3.64	3.74	2.67	4.16	4.53	4.31	4.45	3.03	3.70	3.66	4.29	5.37	5.44	7.96	4.39
10 bits	6.14	4.23	4.49	3.30	3.42	3.11	3.02	2.90	3.32	2.53	2.44	2.83	2.68	4.57	4.56	3.42	3.77	4.03	3.57	3.60
11 bits	6.24	4.23	4.09	3.43	3.02	2.78	2.59	3.17	2.53	2.86	2.87	2.36	2.65	3.01	3.30	3.64	3.33	3.72	3.40	3.33
12 bits	6.24	4.20	4.20	3.71	3.24	3.04	2.66	2.83	2.85	2.81	2.93	2.73	2.61	2.60	2.83	2.93	3.39	3.44	3.69	3.31
13 bits	6.40	4.35	4.33	3.68	3.23	3.22	2.80	3.01	2.94	2.96	2.79	2.88	2.63	2.76	2.62	2.91	2.67	3.15	3.34	3.30
14 bits	6.54	4.54	4.36	4.02	3.43	3.31	2.99	3.17	3.10	3.07	2.92	3.12	2.93	3.06	2.91	3.70	3.32	3.28	3.75	3.55
15 bits	6.68	4.69	4.62	4.06	3.52	3.55	3.25	3.44	3.29	3.26	3.30	3.50	3.36	3.19	3.30	3.71	3.39	3.62	4.04	3.78
16 bits	6.83	4.74	4.66	4.31	3.87	3.70	3.44	3.68	3.63	3.46	3.55	3.64	3.54	3.33	3.69	4.04	4.03	4.01	4.44	4.03
17 bits	6.98	4.94	4.86	4.51	4.02	3.95	3.64	3.88	3.63	3.80	3.74	3.93	3.94	3.73	3.93	4.29	4.38	4.40	4.74	4.28
18 bits	7.13	5.14	4.96	4.56	4.12	4.20	3.79	4.08	3.93	4.00	3.99	4.23	4.29	3.93	4.43	4.43	4.58	4.86	5.23	4.52
32 bits	17.33	15.24	15.51	17.71	17.27	18.60	18.54	20.58	20.93	21.80	23.34	23.88	25.19	25.57	27.67	29.43	28.82	30.90	32.53	22.68

Table C.2: The cost of FIR LPF using SPT_2 number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
8	8.12	6.84	6.63	6.04	3.39	5.37	4.01	3.94	5.02	5.62	6.47	7.29	7.88	8.20	8.20	9.11	7.54	8.12	7.85
9	8.12	6.19	5.29	5.06	4.14	4.39	3.64	2.50	5.56	6.06	5.28	6.20	5.02	4.83	5.17	8.18	7.79	8.18	10.04
10	8.12	6.06	6.07	5.27	4.26	4.48	2.65	2.99	4.96	6.09	5.12	4.29	4.88	6.04	6.10	7.03	6.20	6.92	8.05
11	8.07	6.15	5.79	5.41	3.99	4.48	2.42	3.07	4.92	5.44	5.65	4.25	5.55	5.49	5.27	7.32	7.58	7.54	6.90
12	8.07	6.15	5.79	5.41	4.10	4.48	2.51	2.89	5.05	5.54	5.44	4.39	5.69	5.77	5.76	7.22	7.48	8.17	7.52
14	8.07	6.15	5.79	5.41	4.02	4.48	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.70	5.66	7.50	7.77	8.14	7.65
16	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.70	7.55	7.77	8.19	7.64
18	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.75	7.55	7.77	8.19	7.64

Table C.3: The cost of FIR LPF using SPT_3 number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
8	6.65	6.83	6.41	5.15	3.59	3.75	3.13	3.12	3.38	4.03	5.56	5.07	6.37	6.26	6.26	7.24	6.11	6.77	5.87
9	6.65	5.05	5.24	3.60	4.75	4.09	4.17	3.76	3.90	4.08	4.19	4.20	3.56	4.19	3.61	4.99	4.58	4.51	7.57
10	6.83	5.49	5.71	3.78	4.85	3.28	3.25	3.26	2.96	3.73	3.93	3.44	3.53	4.36	4.37	3.63	3.69	7.35	3.63
11	6.81	5.48	5.55	3.75	4.54	3.31	3.21	3.70	3.32	3.38	3.87	3.16	3.75	2.83	3.29	2.65	2.65	2.08	6.45
12	6.87	5.47	5.45	3.77	4.68	3.43	3.49	3.33	3.90	3.29	3.79	2.99	3.62	3.06	3.01	2.44	3.80	2.30	2.52
14	6.90	5.47	5.45	3.81	4.63	3.45	3.64	3.62	3.84	3.68	3.89	3.30	4.08	3.54	3.38	2.90	3.53	3.12	4.29
16	6.90	5.53	5.55	3.81	4.74	3.54	3.79	3.67	3.89	3.73	3.89	3.40	4.22	3.67	3.77	3.16	3.83	3.30	5.12
18	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.75	7.55	7.77	8.19	7.64

Table C.4: The cost of FIR LPF using SPT_4 number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
8	6.81	5.85	5.83	5.15	3.14	4.13	2.50	3.41	3.14	3.94	5.48	6.39	7.92	7.71	7.71	8.46	7.84	7.71	8.23
9	6.81	4.38	4.15	4.57	4.06	3.84	3.91	2.82	4.36	4.80	4.56	4.60	3.16	3.77	3.73	4.36	5.46	5.52	8.12
10	6.65	4.72	4.84	3.53	3.63	3.38	3.21	3.04	3.55	2.80	3.30	3.02	2.86	4.67	4.67	4.20	3.92	4.88	3.71
11	6.79	4.56	4.36	3.75	3.31	3.19	2.97	3.23	2.87	3.15	3.73	2.69	3.04	3.33	3.42	3.85	4.08	4.64	4.24
12	6.79	4.52	4.47	3.86	3.25	3.48	2.94	2.94	3.12	2.93	3.76	2.74	3.01	3.06	3.10	3.29	4.16	4.68	4.53
14	6.88	4.69	4.49	3.97	3.38	3.62	3.32	3.24	3.31	3.27	3.82	3.23	3.32	3.96	3.59	3.97	4.44	4.95	5.20
16	6.98	4.79	4.64	3.97	3.70	3.75	3.49	3.59	3.58	3.57	4.30	3.66	3.88	4.28	4.48	4.16	5.19	5.92	5.82
18	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.75	7.55	7.77	8.19	7.64

Table C.5: The cost of FIR LPF using $SPTCSD_2$ number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
8	8.12	6.84	6.63	6.04	3.39	5.37	4.01	3.94	5.02	5.62	6.47	7.29	7.88	8.20	8.20	9.11	7.54	8.12	7.85
9	8.12	6.19	5.29	5.06	4.14	4.39	3.64	2.50	5.56	6.06	5.28	6.20	5.02	4.83	5.17	8.18	7.79	8.18	10.04
10	8.12	6.06	6.07	5.27	4.26	4.48	2.65	2.99	4.96	6.09	5.12	4.29	4.88	6.04	6.10	7.03	6.20	6.92	8.05
11	8.07	6.15	5.79	5.41	3.99	4.48	2.42	3.07	4.92	5.44	5.65	4.25	5.55	5.49	5.27	7.32	7.58	7.54	6.90
12	8.07	6.15	5.79	5.41	4.10	4.48	2.51	2.89	4.65	5.54	5.44	4.39	5.69	5.77	5.76	7.22	7.48	8.17	7.52
14	8.07	6.15	5.79	5.41	4.02	4.48	2.59	3.03	5.10	5.62	5.41	4.59	5.69	5.71	5.66	7.52	7.79	8.13	7.65
16	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.75	7.55	7.77	8.19	7.64
18	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.75	7.55	7.77	8.19	7.64

Table C.6: The cost of FIR LPF using $SPTCSD_3$ number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
8	6.65	6.78	6.41	5.15	3.54	3.70	3.08	3.07	3.33	4.03	5.56	5.07	6.37	6.26	6.26	7.19	6.06	6.72	5.82
9	6.65	5.05	5.24	3.60	4.70	4.04	4.12	3.71	3.90	4.03	4.14	4.15	3.51	4.19	3.61	4.99	4.58	4.51	7.52
10	6.83	5.49	5.71	3.78	4.80	3.18	3.20	3.26	2.96	3.68	3.83	3.34	3.48	4.31	4.32	3.58	3.64	7.30	3.58
11	6.81	5.48	5.55	3.75	4.49	3.21	3.21	3.70	2.92	2.83	3.42	3.11	3.70	2.83	3.29	2.60	2.60	2.08	6.40
12	6.87	5.47	5.45	3.77	4.63	3.43	3.49	3.33	3.90	3.24	3.69	2.94	3.52	3.06	3.01	2.34	3.70	2.25	2.47
14	6.90	5.47	5.45	3.81	4.58	3.45	3.64	3.57	3.84	3.58	3.79	3.25	3.88	3.49	3.33	2.85	3.48	3.07	4.19
16	6.90	5.53	5.55	3.81	4.74	3.54	3.79	3.67	3.89	3.73	3.89	3.40	4.22	3.67	3.77	3.11	3.83	3.30	5.12
18	8.07	6.15	5.95	5.41	4.02	4.47	2.59	3.03	5.04	5.63	5.41	4.59	5.77	5.71	5.75	7.55	7.77	8.19	7.64

Table C.7: The cost of FIR LPF using $SPTCSD_4$ number representation

Word Length \ Filter Length	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
8	6.81	5.80	5.83	5.15	3.09	4.08	2.45	3.36	3.09	3.94	5.48	6.39	7.92	7.71	7.71	8.41	7.79	7.66	8.18
9	6.81	4.38	4.15	4.57	4.01	3.79	3.86	2.77	4.26	4.65	4.41	4.55	3.11	3.77	3.73	4.36	5.46	5.52	8.07
10	6.65	4.52	4.79	3.48	3.58	3.23	3.11	2.99	3.40	2.60	3.05	2.87	2.71	4.62	4.62	4.15	3.82	4.78	3.61
11	6.74	4.51	4.36	3.60	3.16	2.99	2.92	3.18	2.17	2.20	3.08	2.39	2.79	3.33	3.42	3.80	4.03	4.49	4.04
12	6.79	4.47	4.47	3.81	3.15	3.28	2.94	2.84	3.02	2.68	3.51	2.44	2.81	3.06	3.10	3.19	4.01	4.48	4.43
14	6.88	4.69	4.49	3.92	3.33	3.52	3.17	3.14	3.16	3.07	3.67	3.03	3.02	3.71	3.49	3.82	4.29	4.75	4.95
16	6.98	4.79	4.64	3.97	3.70	3.75	3.49	3.54	3.48	3.52	4.20	3.61	3.88	4.18	4.38	4.06	5.19	5.82	5.77
18	6.98	4.79	4.64	3.97	3.70	3.75	3.49	3.54	3.48	3.52	4.20	3.61	3.88	4.18	4.38	4.06	5.19	5.82	5.77

REFERENCES

[1] P. John and M. Dimitris, *Digital Signal Processing: Principle,Algorithms, and Applications.* Peason Education, 2007.

[2] A. Ambardar, *Analog And Digtal Signal Processing.* Brooks/Cole, 1999.

[3] "Computer Science 251 Computer Organization," accessed: 2015-1-2. [Online]. Available: http://users.dickinson.edu/ braught/courses/cs251f02/classes/notes07.html

[4] S. Kuo and B. Lee, *Real-Time Digital Signal Processing.* Wiley, 2001.

[5] T. Parks, *Digtal Filter Design.* Wiley, 1987.

[6] Y.J.Yu and Y.C.Lim, "Roundoff noise analysis of signals represented using signed power-of-two terms," in *Signal Processing Conference, 2006 14th European*, Sept 2006, pp. 1–4.

[7] S. Chrisomalis, *Numerical Notation: A Comparative History.* Cambridge University Press, 2010.

[8] "Binary number," accessed: 2015-4-2. [Online]. Available: http://https://en.wikipedia.org/wiki/Binarynumber

[9] "Representation of Numbers," accessed: 2015-10-10. [Online]. Available: http://www.swarthmore.edu/NatSci/echeeve1/Ref/BinaryMath/NumSys.html

[10] N. Bose, *Digital Filter Theory and Applications.* Elsevier Science, 1985.

[11] P. Naidu, *Modern Digital Signal Processing.* Alpha Science, 2006.

[12] T. Elali, *Discrete Systems and Digital Signal Processing with Matlab.* CRC Press, 2003.

[13] "FIR Filter Properties," accessed: 2015-5-28. [Online]. Available: http://dspguru.com/dsp/faqs/fir/properties

[14] R. Lyons, *Understand Digital Signal Processing.* Bernard Goodwin, 2004.

[15] "Pole-Zero plot - Theory/Equations," accessed: 2015-9-21. [Online]. Available: https://www.ee.columbia.edu/ dpwe/e4810/matlab/pezdemo/help/theory.html

[16] "FIR Filter Implementation," accessed: 2015-7-2. [Online]. Available: http://dspguru.com/dsp/faqs/fir/implementation

[17] D. Chan and L. Rabiner, "Analysis of quantization errors in the direct form for finite impulse response digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 4, pp. 354–366, Aug 1973.

[18] Y. J. Yu and Y. C. Lim, "Roundoff noise analysis of signals represented using signed power-of-two terms," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2122–2135, May 2007.

[19] S. Mitra, K. Hirano, and H. Sakaguchi, "A simple method of computing the input quantization and multiplication roundoff errors in a digital filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 5, pp. 326–329, Oct 1974.

[20] A. Sripad and D. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 442–448, Oct 1977.

[21] J. Knowles and E. Olcayto, "Coefficient accuracy and digital filter response," *IEEE Transactions on Circuit Theory*, vol. 15, no. 1, pp. 31–41, Mar 1968.

[22] M. Press, *Coefficient Accuracy and Digital Filter Response.* MIT Press, 1969, pp. 103–113. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6301532

[23] J. J. Nielsen, "Design of linear-phase direct-form fir digital filters with quantized coefficients using error spectrum shaping," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 1020–1026, Jul 1989.

[24] J. P. Dugre, A. Beex, and L. Scharf, "Generating covariance sequences and the calculation of quantization and rounding error variances in digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 102–104, Feb 1980.

[25] I. Koren, *Computer arithmetic algorithms*. Universities Press.

[26] L. Franz, Booth.A.D, and Meagher.R.E, *Advances in computers*. Academic Press.

[27] R. Hartley, "Optimization of canonic signed digit multipliers for filter design," in *Circuits and Systems, 1991., IEEE International Sympoisum on*, Jun 1991, pp. 1992–1995 vol.4.

[28] D. S. Phatak and I. Koren, "Hybrid signed-digit number systems: a unified framework for redundant number representations with bounded carry propagation chains," *IEEE Transactions on Computers*, vol. 43, no. 8, pp. 880–891, Aug 1994.

[29] B. Parhami, "Generalized signed-digit number systems: a unifying framework for redundant number representations," *IEEE Transactions on Computers*, vol. 39, no. 1, pp. 89–98, Jan 1990.

[30] "Fast Arithmetic on FPGA Using Redundant Binary Apparatus," http://www.louif.com/rbin/, accessed: 2015-10-30.

[31] F. Al-Hasani, M. Hayes, and A. Bainbridge-Smith, "A new subexpression elimination algorithm using zero-dominant set," in *Electronic Design, Test and Application (DELTA), 2011 Sixth IEEE International Symposium on*, Jan 2011, pp. 45–50.

[32] F. Al Hasani, M. Hayes, and A. Bainbridge-Smith, "A common subexpression elimination tree algorithm," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 9, pp. 2389–2400, Sept 2013.

[33] V. Sorokine and S. Pasupathy, "On the hamming weight of binary sequences and linear complexity," in *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, Aug 1998, pp. 103–.

[34] R. Hewlitt and J. Swartzlantler, E.S., "Canonical signed digit representation for fir digital filters," in *Signal Processing Systems, 2000. SiPS 2000. 2000 IEEE Workshop on*, 2000, pp. 416–426.

[35] M. A. Soderstrand, "Csd multipliers for fpga dsp applications," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 5, May 2003, pp. V–469–V–472 vol.5.

[36] K.-Y. Khoo, A. Kwentus, and J. Willson, A.N., "A programmable fir digital filter using csd coefficients," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 6, pp. 869–874, Jun 1996.

[37] K. Suzuki, H. Ochi, and S. Kinjo, "A design of fir filter using csd with minimum number of registers," in *Circuits and Systems, 1996., IEEE Asia Pacific Conference on*, Nov 1996, pp. 227–230.

[38] M. Yamada and A. Nishihara, "High-speed fir digital filter with csd coefficients implemented on fpga," in *Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific*, 2001, pp. 7–8.

[39] L. S. DeBrunner, V. E. DeBrunner, and D. Bhogaraju, "Defining canonical-signed-digit number systems as arithmetic codes," in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, vol. 2, Nov 2002, pp. 1593–1597 vol.2.

[40] M. Tanaka and A. Nishihara, "Design of signal word decomposed filters with canonical-signed digit coefficients," in *TENCON 2000. Proceedings*, vol. 1, 2000, pp. 482–486 vol.1.

[41] K. KeiYong, A. Kwentus, and J. Willson, A.N., "A programmable fir digital filter using csd coefficients," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 6, pp. 869–874, Jun 1996.

[42] X. Fei, C. Chip-Hong, and J. Ching-Chuen, "Hamming weight pyramid a new insight into canonical signed digit representation and its applications," *Computers and Electrical Engineering*, vol. 33, no. 3, pp. 195 – 207, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790606001005

[43] C.-Y. Yao, "A study of spt-term distribution of csd numbers and its application for designing fixed-point linear phase fir filters," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 2, May 2001, pp. 301–304 vol. 2.

[44] Y. C. Lim, R. Yang, D. Li, and J. Song, "Signed power-of-two term allocation scheme for the design of digital filters," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 46, no. 5, pp. 577–584, May 1999.

[45] "A polynomial-time algorithm for designing digital filters with power-of-two coefficients," in *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, May 1993, pp. 84–87.

[46] Y. Lim and S. Parker, "Fir filter design over a discrete powers-of-two coefficient space," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, no. 3, pp. 583–591, Jun 1983.

[47] O. Gustafsson, H. Johansson, and L. Wanhammar, "Milp design of frequency-response masking fir filters with few spt terms," in *Control, Communications and Signal Processing, 2004. First International Symposium on*, 2004, pp. 405–408.

[48] C.-L. Chen and A. N. Willson, "A trellis search algorithm for the design of fir filters with signed-powers-of-two coefficients," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 1, pp. 29–39, Jan 1999.

[49] E. Eweda, "Convergence analysis and design of an adaptive filter with finite-bit power-of-two quantized error," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 113–115, Feb 1992.

[50] W. Lu, "Design of 2-d fir filters with power-of-two coefficients: a semidefinite programming relaxation approach," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 2, May 2001, pp. 549–552 vol. 2.

[51] Y. C. Lim and Y. J. Yu, "A width-recursive depth-first tree search approach for the design of discrete coefficient perfect reconstruction lattice filter bank," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 6, pp. 257–266, June 2003.

[52] D. Rao, "Analysis of coefficient quantization errors in state-space digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 131–139, Feb 1986.

[53] Y. C. Lim, "Design of discrete-coefficient-value linear phase fir filters with optimum normalized peak ripple magnitude," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 12, pp. 1480–1486, Dec 1990.